The Bro Monitoring Platform

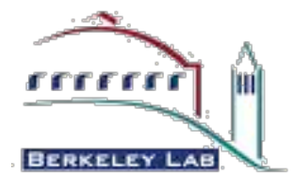
Robin Sommer

International Computer Science Institute, &
Lawrence Berkeley National Laboratory

robin@icsi.berkeley.edu
http://www.icir.org/robin

"What Is Bro?"


ICSI
INTERNATIONAL
COMPUTER SCIENCE
INSTITUTE


BERKELEY LAB

"What Is Bro?"

Packet Capture

"What Is Bro?"

Packet Capture

Traffic Inspection

"What Is Bro?"

Packet Capture

Traffic Inspection

Attack Detection

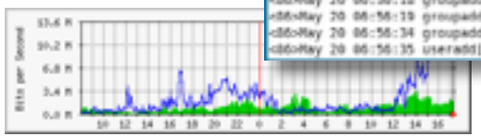
"What Is Bro?"

Packet Capture

Traffic Inspection

Attack Detection

NetFlow



syslog

Log Recording

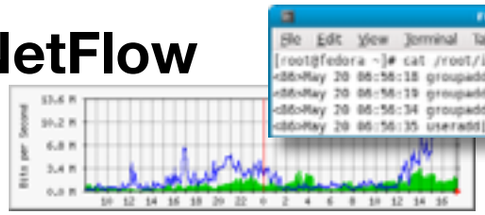
"What Is Bro?"

Packet Capture

Traffic Inspection

Attack Detection

NetFlow



syslog

Log Recording

Flexibility
Abstraction
Data Structures

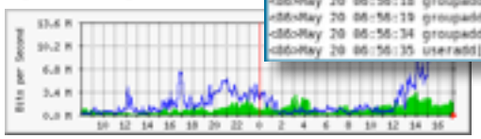
"What Is Bro?"

Packet Capture

Traffic Inspection

Attack Detection

NetFlow



syslog

Log Recording

Flexibility
Abstraction
Data Structures

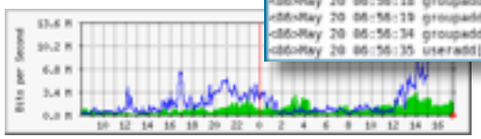
"What Is Bro?"

Packet Capture

Traffic Inspection

Attack Detection

NetFlow



syslog

Log Recording

Flexibility
Abstraction
Data Structures

"What Is Bro?"

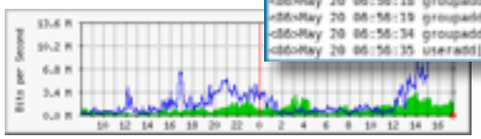
Packet Capture

Traffic Inspection

Attack Detection

"Domain-specific Python"

NetFlow

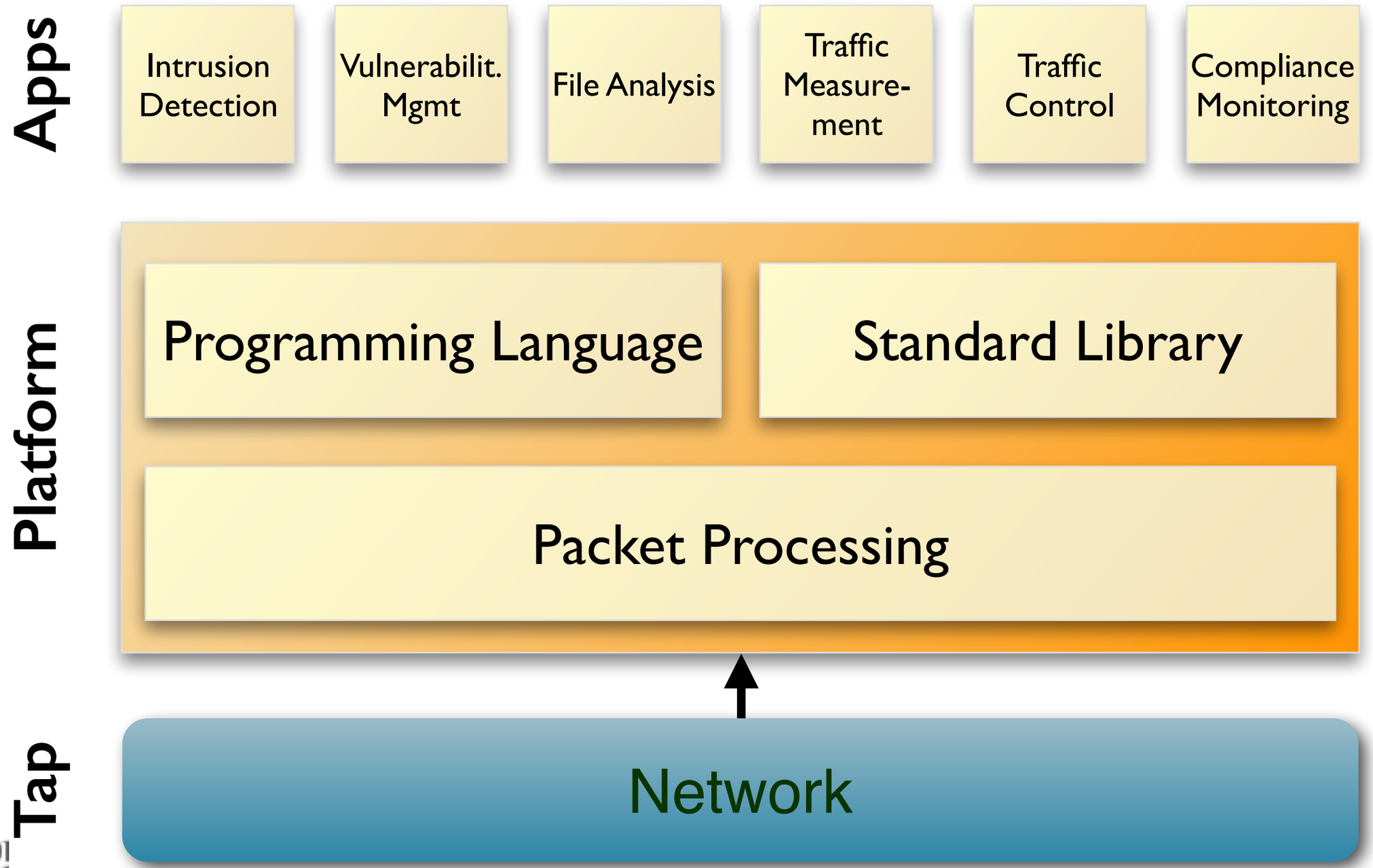


syslog

Log Recording

Flexibility
Abstraction
Data Structures

The Bro Platform



The Bro Platform

Open Source
BSD License

Apps

- Intrusion Detection
- Vulnerabilit. Mgmt
- File Analysis
- Traffic Measure-ment
- Traffic Control
- Compliance Monitoring

Platform

- Programming Language
- Standard Library
- Packet Processing

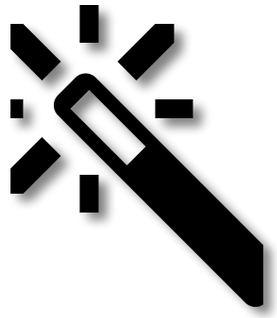
Tap

- Network

"What Can It Do?"

Log Files

Alerts

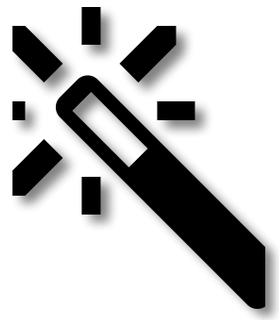


Custom Logic

"What Can It Do?"

"Network Ground Truth"

Alerts



Custom Logic

Bro Logs

```
> bro -i en0
[ … wait … ]
```

Bro Logs

```
> bro -i en0
[ … wait … ]
> ls *.log
app_stats.log          irc.log                socks.log
communication.log      known_certs.log        software.log
conn.log               known_hosts.log        ssh.log
dhcp.log               known_services.log     ssl.log
dns.log                modbus.log             syslog.log
dpd.log                notice.log             traceroute.log
files.log              reporter.log           tunnel.log
ftp.log                signatures.log         weird.log
http.log               smtp.log
```

Bro Logs

```
> bro -i en0
[ … wait … ]
> cat conn.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2013-04-28-23-47-26
#fields ts          uid             id.orig_h       id.orig_p  id.resp_h       […]
#types  time        string          addr            port       addr            […]
1258531221.486539  arKYeMETxOg     192.168.1.102   68         192.168.1.1     […]
1258531680.237254  nQcgTWjvg4c     192.168.1.103   37         192.168.1.255   […]
1258531693.816224  j4u32Pc5bif     192.168.1.102   37         192.168.1.255   […]
1258531635.800933  k6kgXLOoSKl     192.168.1.103   138        192.168.1.255   […]
1258531693.825212  TEfuqmmG4bh     192.168.1.102   138        192.168.1.255   […]
1258531803.872834  5OKnoww6xl4     192.168.1.104   137        192.168.1.255   […]
1258531747.077012  FrJExwHcSal     192.168.1.104   138        192.168.1.255   […]
1258531924.321413  3PKsZ2Uye21     192.168.1.103   68         192.168.1.1     […]
[…]
```

Connections Logs

conn.log

ts	**1393099191.817686**	Timestamp
uid	**Cy3S2U2sbarorQgmw6a**	Unique ID
id.orig_h	**177.22.211.144**	Originator IP
id.orig_p	**43618**	Originator Port
id.resp_h	**115.25.19.26**	Responder IP
id.resp_p	**25**	Responder Port
proto	**tcp**	IP Protocol
service	**smtp**	App-layer Protocol
duration	**1.414936**	Duration
orig_bytes	**9068**	Bytes by Originator
resp_bytes	**4450**	Bytes by Responder
conn_state	**SF**	TCP state
local_orig	**T**	Local Originator?
missed_bytes	**0**	Gaps
history	**ShAdDaFf**	State History
tunnel_parents	**(empty)**	Outer Tunnels

HTTP

http.log

ts	1393099291.589208
uid	CKFUW73bIADw0r9pl
id.orig_h	17.22.7.4
id.orig_p	54352
id.resp_h	24.26.13.36
id.resp_p	80
method	POST
host	com-services.pandonetworks.com
uri	/soapservices/services/SessionStart
referrer	-
user_agent	Mozilla/4.0 (Windows; U) Pando/2.6.0.8
status_code	200
username	anonymous
password	-
orig_mime_types	application/xml
resp_mime_types	application/xml

SSL

ssl.log

ts	1392805957.927087
uid	CEA05l2D7k0BD9Dda2
id.orig_h	2a07:f2c0:90:402:41e:c13:6cb:99c
id.orig_p	40475
id.resp_h	2406:fe60:f47::aaeb:98c
id.resp_p	443
version	TLSv10
cipher	TLS_DHE_RSA_WITH_AES_256_CBC_SHA
server_name	www.netflix.com
subject	CN=www.netflix.com,OU=Operations, O=Netflix, Inc.,L=Los Gatos, ST=CALIFORNIA,C=US
issuer_subject	CN=VeriSign Class 3 Secure Server CA, OU=VeriSign Trust Network,O=VeriSign, C=US
not_valid_before	1389859200.000000
not_valid_after	1452931199.000000
client_subject	-
client_issuer_subject	-
cert_hash	197cab7c6c92a0b9ac5f37cfb0699268
validation_status	ok

Syslog & DHCP

syslog.log

ts	1392796803.311801
uid	CnYivt3ZONHOuBALR8
id.orig_h	12.3.8.161
id.orig_p	514
id.resp_h	16.74.12.24
id.resp_p	514
proto	udp
facility	AUTHPRIV
severity	INFO
message	sshd[13825]: Accepted publickey for harvest from xxx.xxx.xxx.xxx

Syslog & DHCP

syslog.log

ts	1392796803.311801
uid	CnYivt3ZONHOuBALR8
id.orig_h	12.3.8.161
id.orig_p	514
id.resp_h	16.74.12.24
id.resp_p	514
proto	udp
facility	AUTHPRIV
severity	INFO
message	sshd[13825]: Accepted publickey for harvest from xxx.xxx.xxx.xxx

dhcp.log

ts	1392796962.091566
uid	Ci3RM24iF4vIYRGHc3
id.orig_h	10.129.5.11
id.resp_h	10.129.5.1
mac	04:12:38:65:fa:68
assigned_ip	10.129.5.11
lease_time	14400.000000

Files

files.log

ts	1392797643.447056
fuid	FnungQ3TI19GahPJP2
tx_hosts	191.168.187.33
rx_hosts	10.1.29.110
conn_uids	CbDgik2fjeKL5qzn55
source	SMTP
analyzers	SHA1,MD5
mime_type	application/x-dosexec
filename	Letter.exe
duration	5.320822
local_orig	T
seen_bytes	39508
md5	93f7f5e7a2096927e06e[…]1085bfcfb
sha1	daed94a5662a920041be[…]a433e501646ef6a03
extracted	-

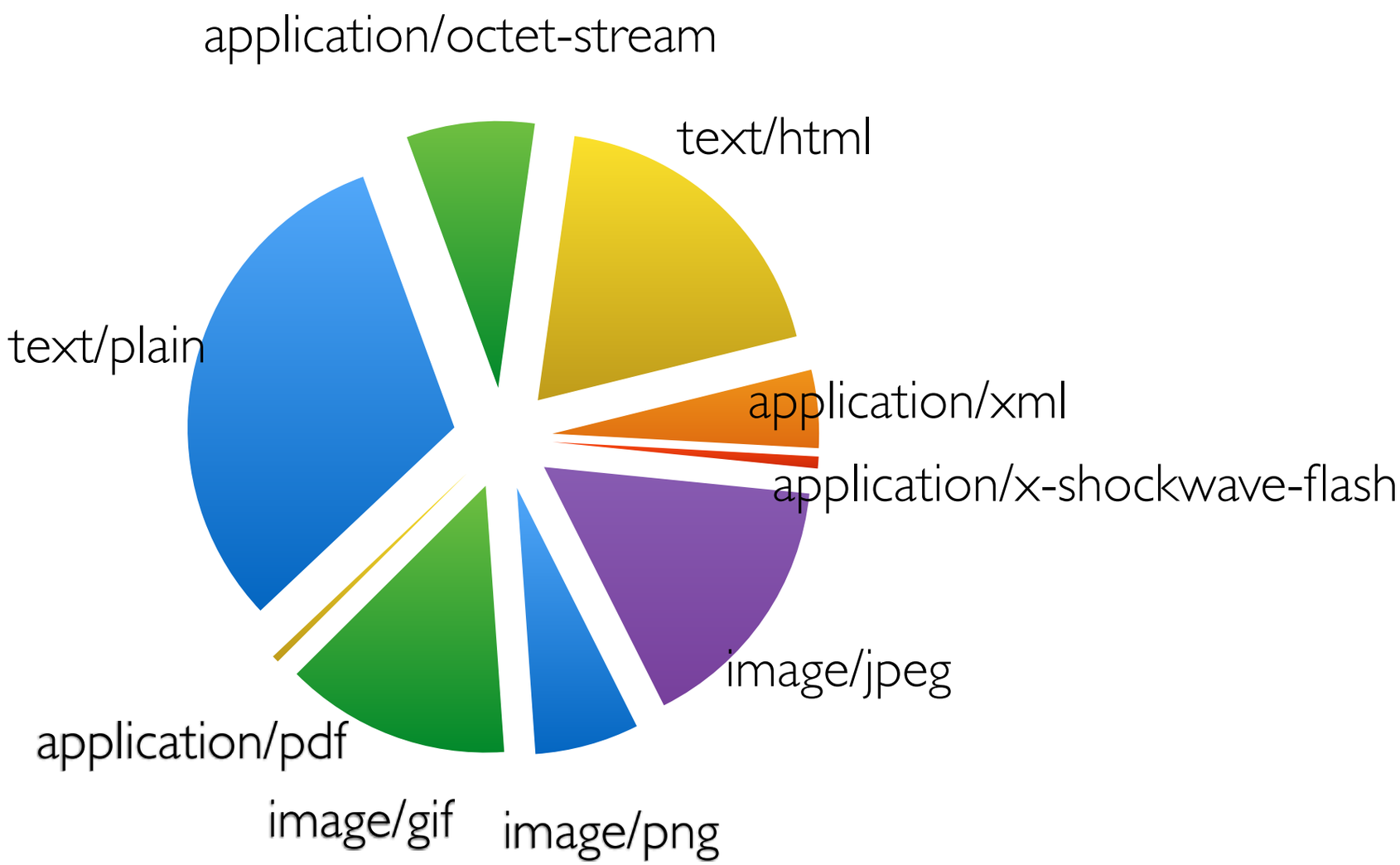
Software

software.log

ts	1392796839.675867
host	10.209.100.2
host_p	-
software_type	HTTP::BROWSER
name	DropboxDesktopClient
version.major	2
version.minor	4
version.minor2	11
version.minor3	-
version.addl	Windows
unparsed_version	DropboxDesktopClient/2.4.11 (Windows; 8; i32; en_US)

Help Understand Your Network

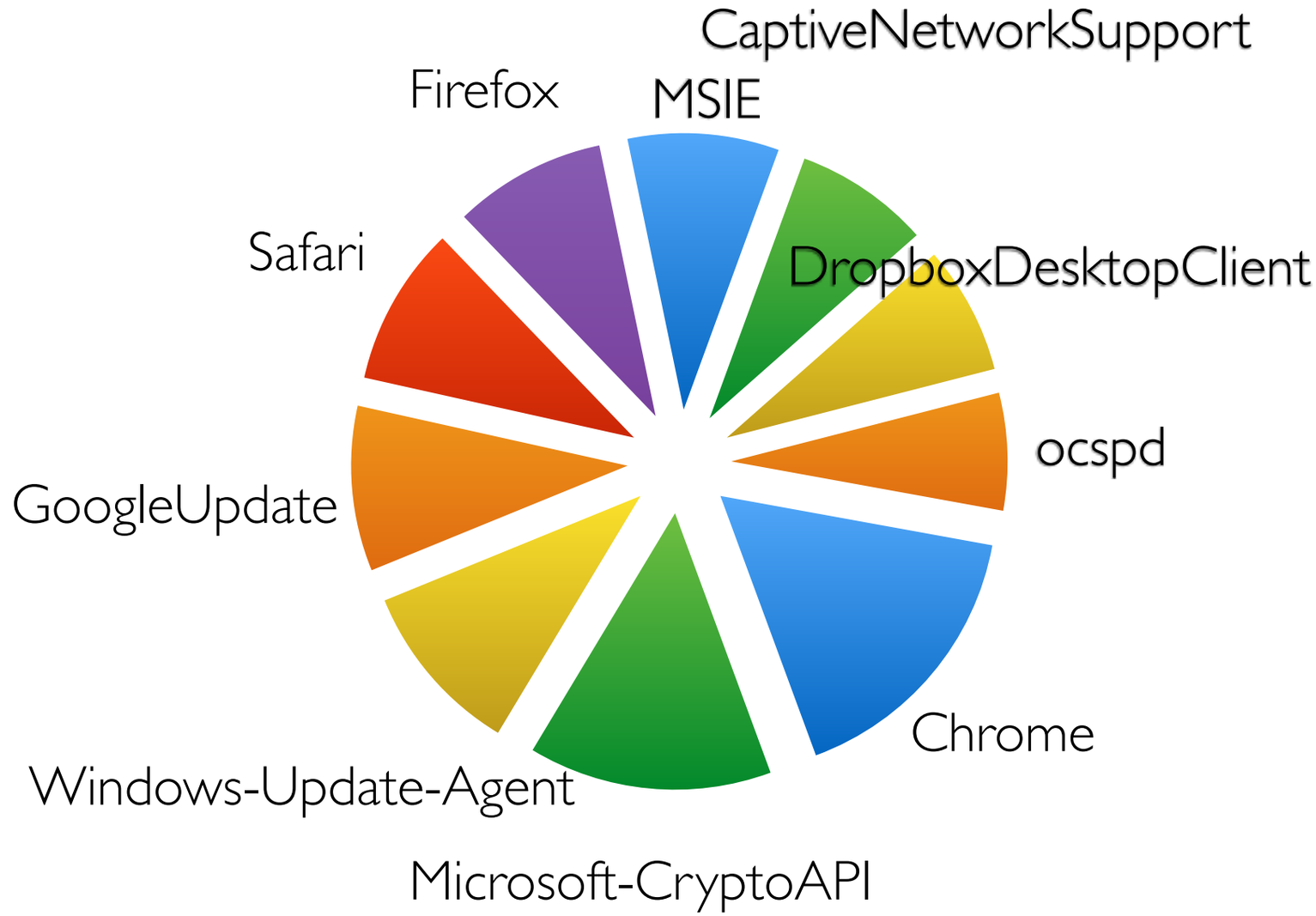
Top File Types




```
cat files.log | bro-cut mime_type | sort | uniq -c | sort -rn
```

Help Understand Your Network (2)

Top Software by Number of Hosts

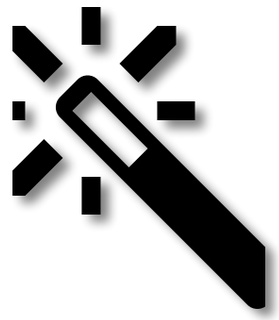



```
cat software.log  | bro-cut host name | sort | uniq |
 awk -F '\t' '{print $2}' | sort | uniq -c | sort -rn
```

"What Can It Do?"

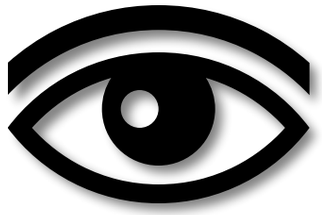
Log Files

Alerts

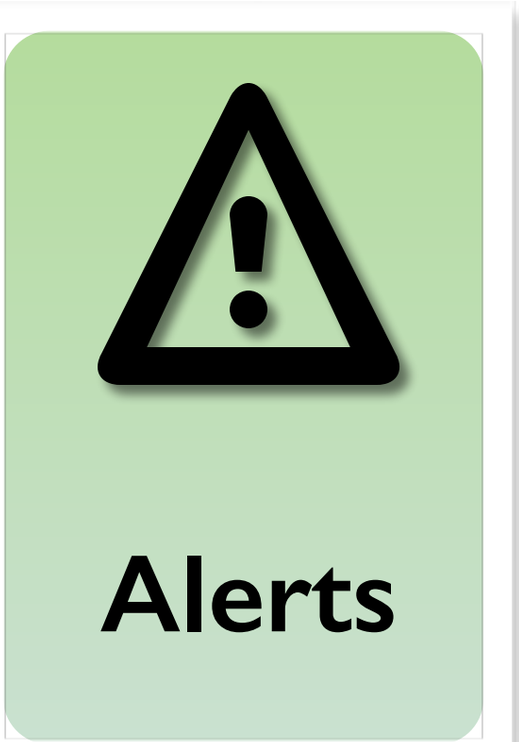


Custom Logic

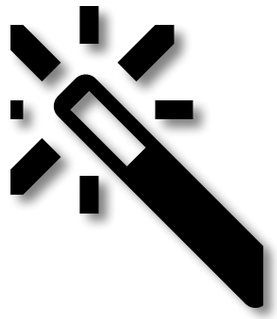
"What Can It Do?"



Log Files



Alerts



Custom Logic

"Watch this!"

Recorded in notice.log.
Can trigger actions.

Alerts

```
CaptureLoss::Too_Much_Loss
Conn::Ack_Above_Hole
Conn::Content_Gap
Conn::Retransmission_Inconsistency
DNS::External_Name
FTP::Bruteforcing
FTP::Site_Exec_Success
HTTP::SQL_Injection_Attacker
HTTP::SQL_Injection_Victim
Intel::Notice
PacketFilter::Dropped_Packets
ProtocolDetector::Protocol_Found
ProtocolDetector::Server_Found
SMTP::Blocklist_Blocked_Host
SMTP::Blocklist_Error_Message
SMTP::Suspicious_Origination
SSH::Interesting_Hostname_Login
SSH::Login_By_Password_Guesser
SSH::Password_Guessing
SSH::Watched_Country_Login
SSL::Certificate_Expired
SSL::Certificate_Expires_Soon
SSL::Certificate_Not_Valid_Yet
SSL::Invalid_Server_Cert
Scan::Address_Scan
Scan::Port_Scan
Signatures::Count_Signature
Signatures::Multiple_Sig_Responders
Signatures::Multiple_Signatures
Signatures::Sensitive_Signature
Software::Software_Version_Change
Software::Vulnerable_Version
TeamCymruMalwareHashRegistry::Match
Traceroute::Detected
Weird::Activity
```

Watching for Suspicious Logins

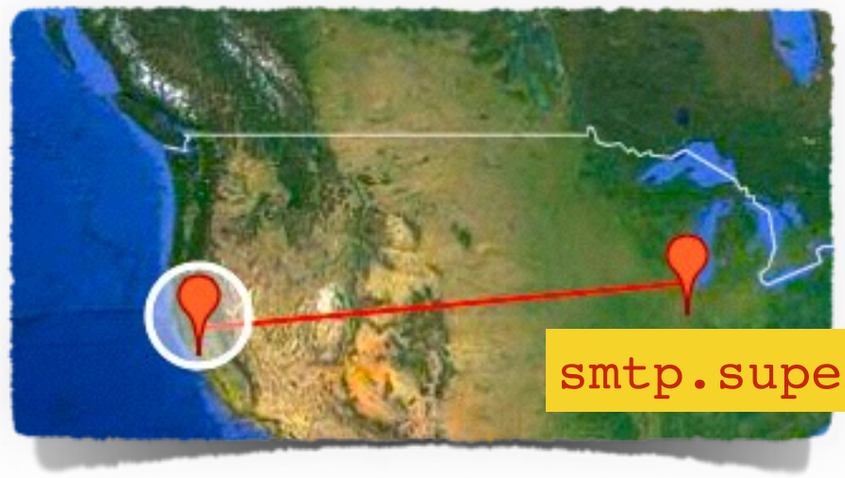
`SSH::Watched_Country_Login`

Login from an unexpected country.

Watching for Suspicious Logins

`SSH::Watched_Country_Login`

Login from an unexpected country.

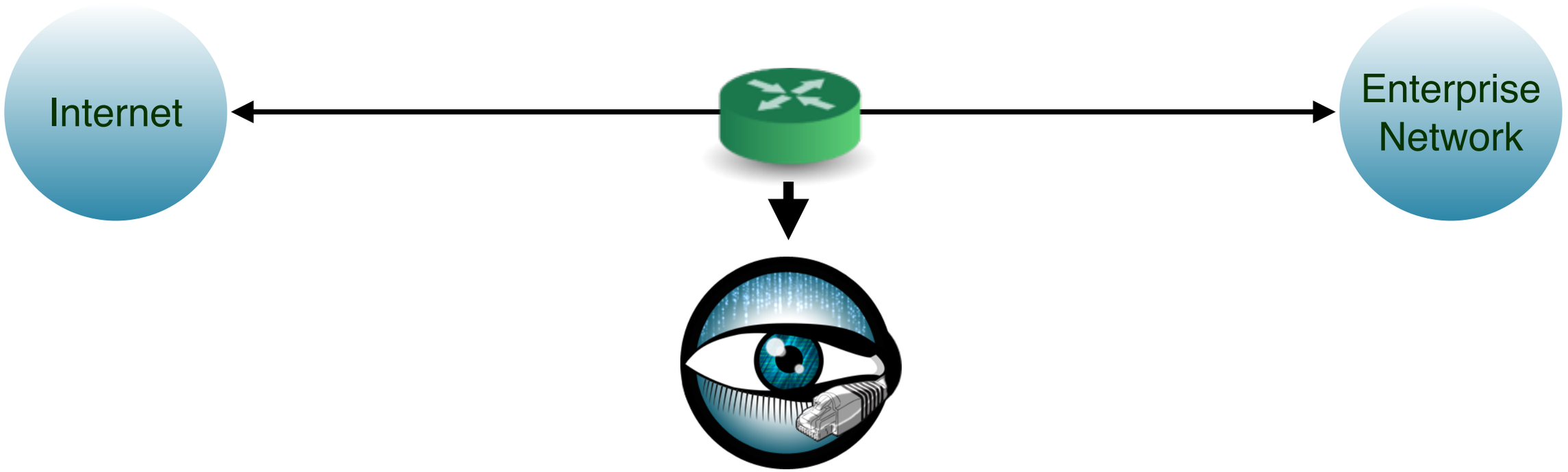


`SSH::Interesting_Hostname_Login`

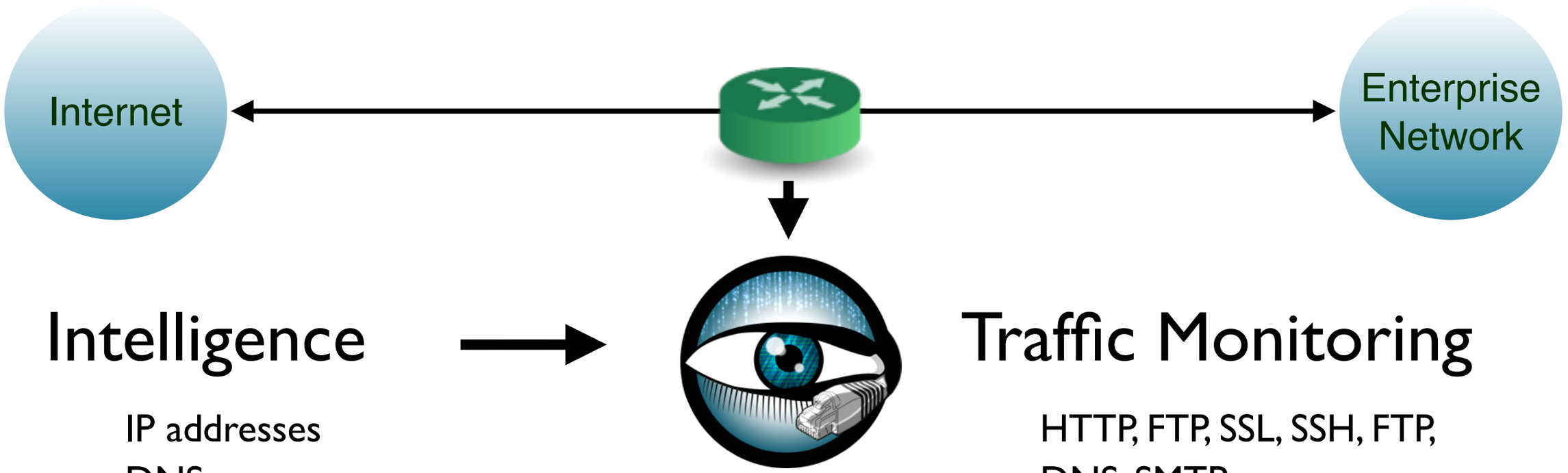
Login from an unusual host name.

`smtp.supercomputer.edu`

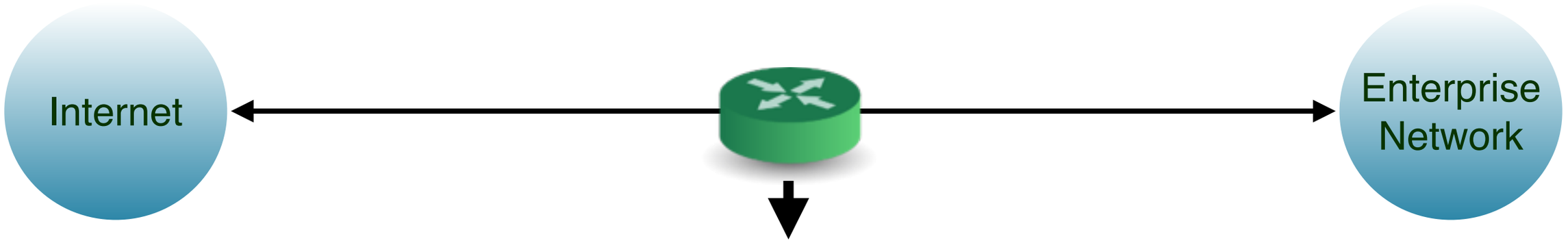
Intelligence Integration (Passive)



Intelligence Integration (Passive)



Intelligence Integration (Passive)



Intelligence

IP addresses
DNS names
URLs
File hashes

Feeds

CIF
JC3
Spamhaus
Custom/Proprietary

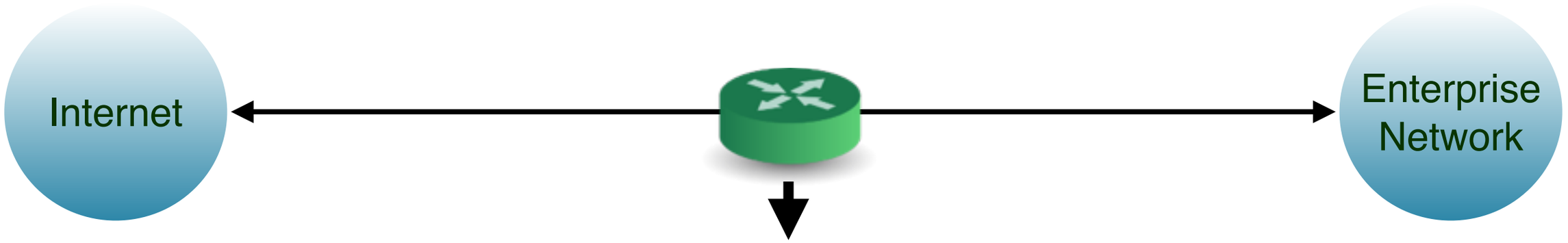
Traffic Monitoring

HTTP, FTP, SSL, SSH, FTP,
DNS, SMTP, …

ts	**1258565309.806483**
uid	**CAK677xaOmi66X4Th**
id.orig_h	**192.168.1.103**
id.resp_h	**192.168.1.1**
note	**Intel::Notice**
indicator	**baddomain.com**
indicator_type	**Intel::DOMAIN**
where	**HTTP::IN_HOST_HEADER**
source	**My-Private-Feed**

notice.log

Intelligence Integration (Passive)




```
Conn::IN_ORIG
Conn::IN_RESP
Files::IN_HASH
Files::IN_NAME
DNS::IN_REQUEST
DNS::IN_RESPONSE
HTTP::IN_HOST_HEADER
HTTP::IN_REFERRER_HEADER
HTTP::IN_USER_AGENT_HEADER
HTTP::IN_X_FORWARDED_FOR_HEADER
HTTP::IN_URL
SMTP::IN_MAIL_FROM
SMTP::IN_RCPT_TO
SMTP::IN_FROM
SMTP::IN_TO
SMTP::IN_RECEIVED_HEADER
SMTP::IN_REPLY_TO
SMTP::IN_X_ORIGINATING_IP_HEADER
SMTP::IN_MESSAGE
SSL::IN_SERVER_CERT
SSL::IN_CLIENT_CERT
SSL::IN_SERVER_NAME
SMTP::IN_HEADER
```

Traffic Monitoring

HTTP, FTP, SSL, SSH, FTP,
DNS, SMTP, …

ts	1258565309.806483
uid	CAK677xaOmi66X4Th
id.orig_h	192.168.1.103
id.resp_h	192.168.1.1
note	Intel::Notice
indicator	baddomain.com
indicator_type	Intel::DOMAIN
where	HTTP::IN_HOST_HEADER
source	My-Private-Feed

notice.log

Intelligence Integration (Active)

Intelligence Integration (Active)


```
# cat files.log | bro-cut mime_type sha1 | awk '$1 ~ /x-dosexec/‘
application/x-dosexec   5fd2f37735953427e2f6c593d6ec7ae882c9ab54
application/x-dosexec   00c69013d34601c2174b72c9249a0063959da93a
application/x-dosexec   0d801726d49377bfe989dcca7753a62549f1ddda
[…]
```

Intelligence Integration (Active)


```
# cat files.log | bro-cut mime_type sha1 | awk '$1 ~ /x-dosexec/‘
application/x-dosexec   5fd2f37735953427e2f6c593d6ec7ae882c9ab54
application/x-dosexec   00c69013d34601c2174b72c9249a0063959da93a
application/x-dosexec   0d801726d49377bfe989dcca7753a62549f1ddda
[…]
```

```
# dig +short 733a48a9cb4[…]2a91e8d00.malware.hash.cymru.com TXT
"1221154281 53"
```

Intelligence Integration (Active)

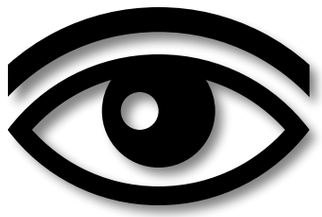

```
# cat files.log | bro-cut mime_type sha1 | awk '$1 ~ /x-dosexec/‘
application/x-dosexec   5fd2f37735953427e2f6c593d6ec7ae882c9ab54
application/x-dosexec   00c69013d34601c2174b72c9249a0063959da93a
application/x-dosexec   0d801726d49377bfe989dcca7753a62549f1ddda
[…]
```

```
# dig +short 733a48a9cb4[…]2a91e8d00.malware.hash.cymru.com TXT
"1221154281 53"
```

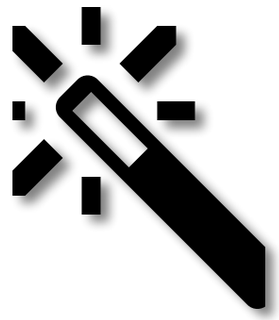
notice.log

ts	**1392423980.736470**	Timestamp
uid	**CjKeSB45xaOmiIo4Th**	Connection ID
id.orig_h	**10.2.55.3**	Originator IP
id.resp_h	**192.168.34.12**	Responder IP
fuid	**FEGVbAgcArRQ49347**	File ID
mime_type	**application/jar**	MIME type
description	**http://app.looking3g.com/[…]**	Source URL Bro saw
note	**TeamCymruMalwareHashRegistry::Match**	Notice Type
msg	**2013-09-14 22:06:51 / 20%**	MHR reply
sub	**https://www.virustotal.com/[…]**	VirusTotal URL

"What Can It Do?"

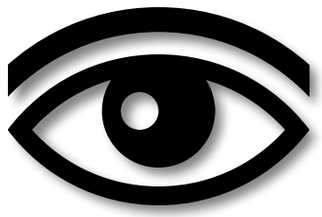


Log Files



Custom Logic

"What Can It Do?"

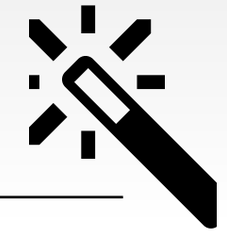


Log Files

Alerts

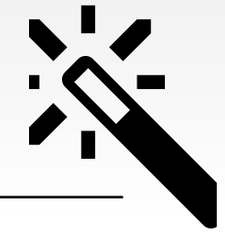
"Don't ask what Bro can do.
*Ask what **you want** it to do."*

Script Example: Matching URLs



Task: Report all Web requests for files called "passwd".

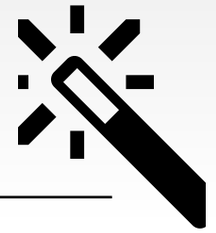
Script Example: Matching URLs



Task: Report all Web requests for files called "passwd".

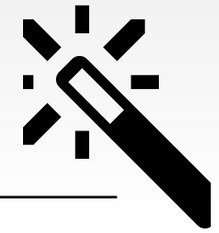
```
event http_request(c: connection,           # Connection.
                   method: string,          # HTTP method.
                   original_URI: string,    # Requested URL.
                   unescaped_URI: string, # Decoded URL.
                   version: string)         # HTTP version.
{
    if ( method == "GET" && unescaped_URI == /.*passwd/ )
        NOTICE(...); # Alarm.
}
```

Script Example: Scan Detector



Task: Count failed connection attempts per source address.

Script Example: Scan Detector



Task: Count failed connection attempts per source address.

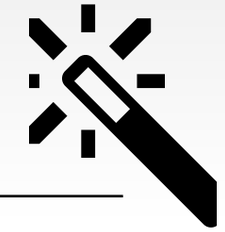
```
global attempts: table[addr] of count &default=0;

event connection_rejected(c: connection)
{
    local source = c$id$orig_h;      # Get source address.

    local n = ++attempts[source];    # Increase counter.

    if ( n == SOME_THRESHOLD )       # Check for threshold.
        NOTICE(...);                 # Alarm.
}
```

Scripts are Bro's "Magic Ingredient"



Bro comes with >10,000 lines of script code.

Prewritten functionality that's just loaded.

Scripts generate everything we have seen.

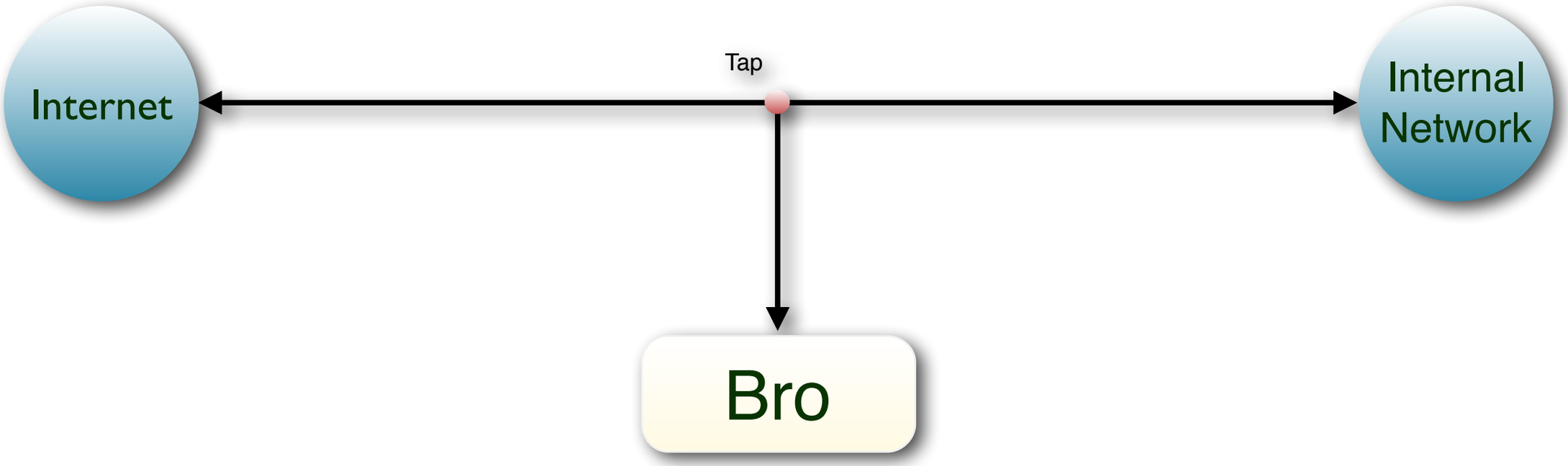
Amendable to extensive customization and extension.

Growing community writing 3rd party scripts.

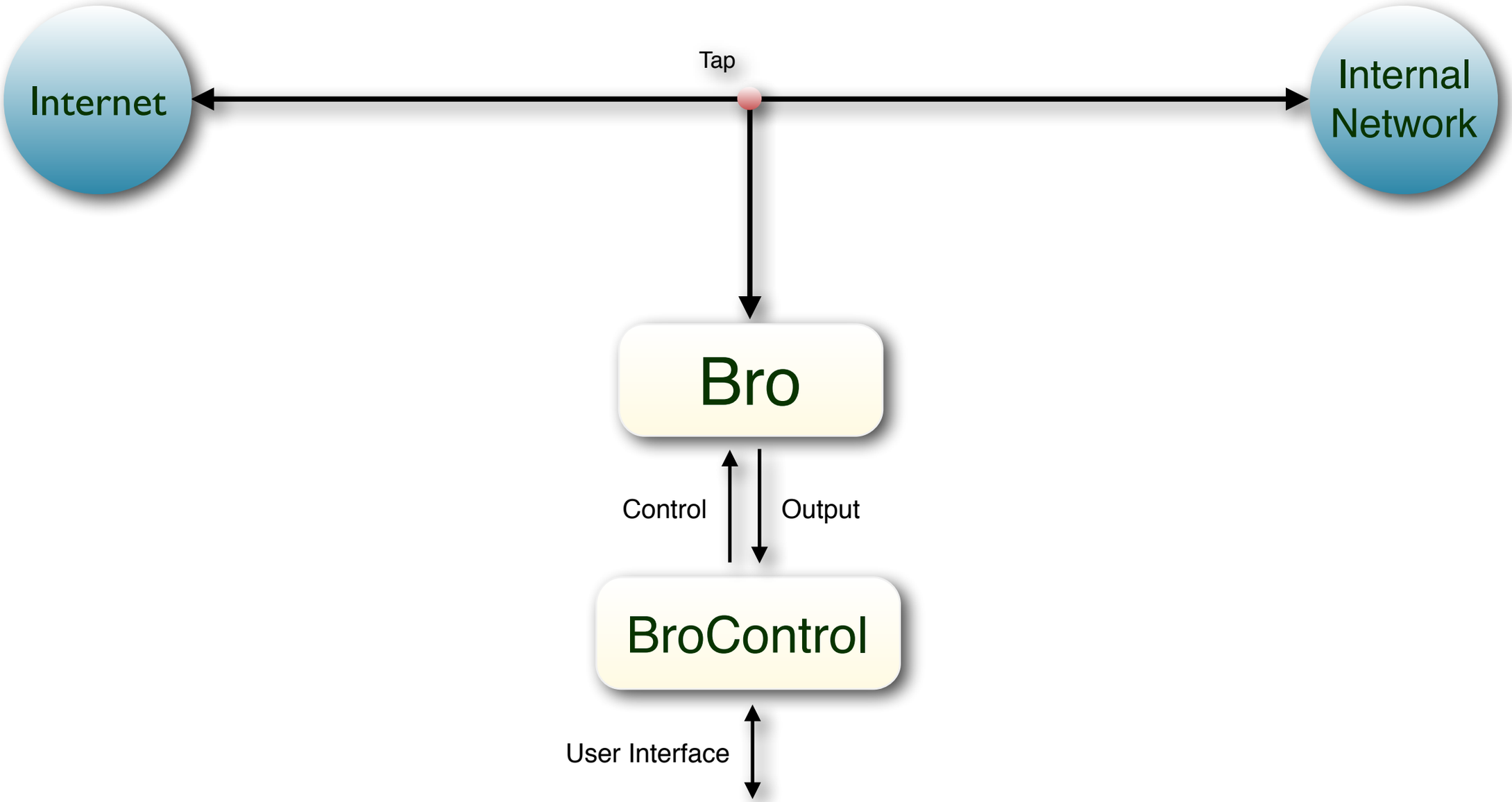
Bro could report Mandiant's APT1 indicators within a day.

Bro Ecosystem

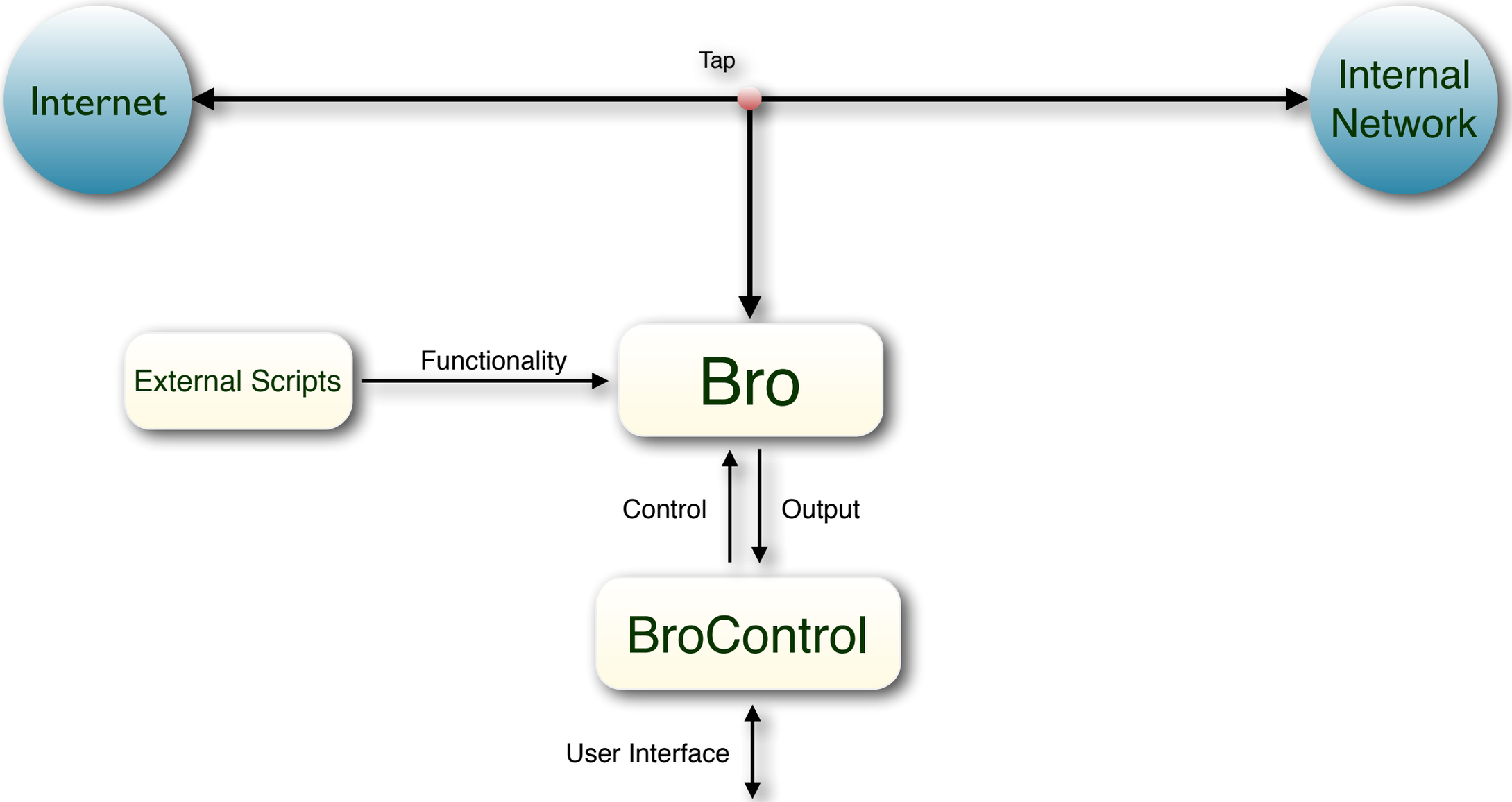
Bro Ecosystem



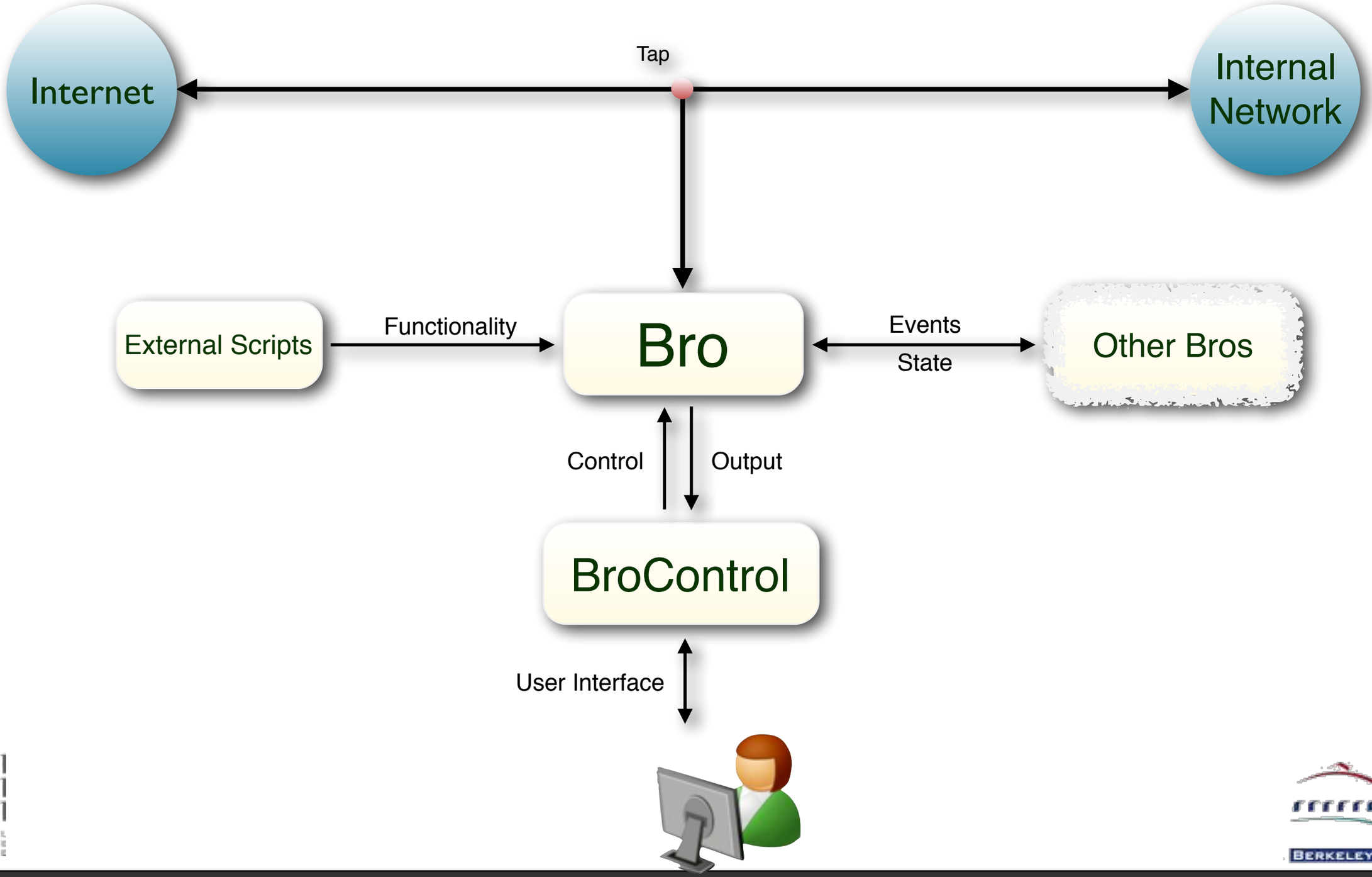
Bro Ecosystem



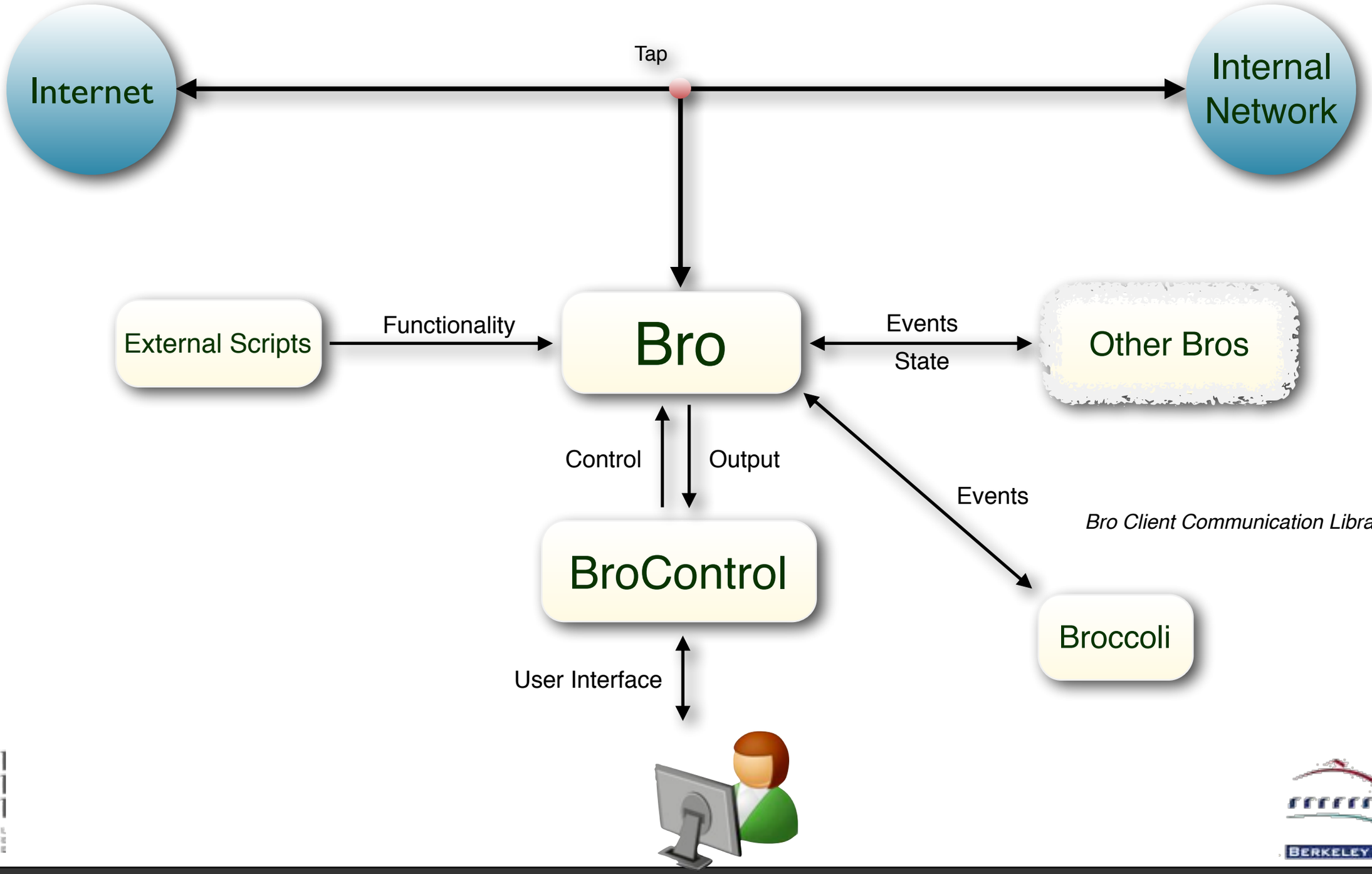
Bro Ecosystem



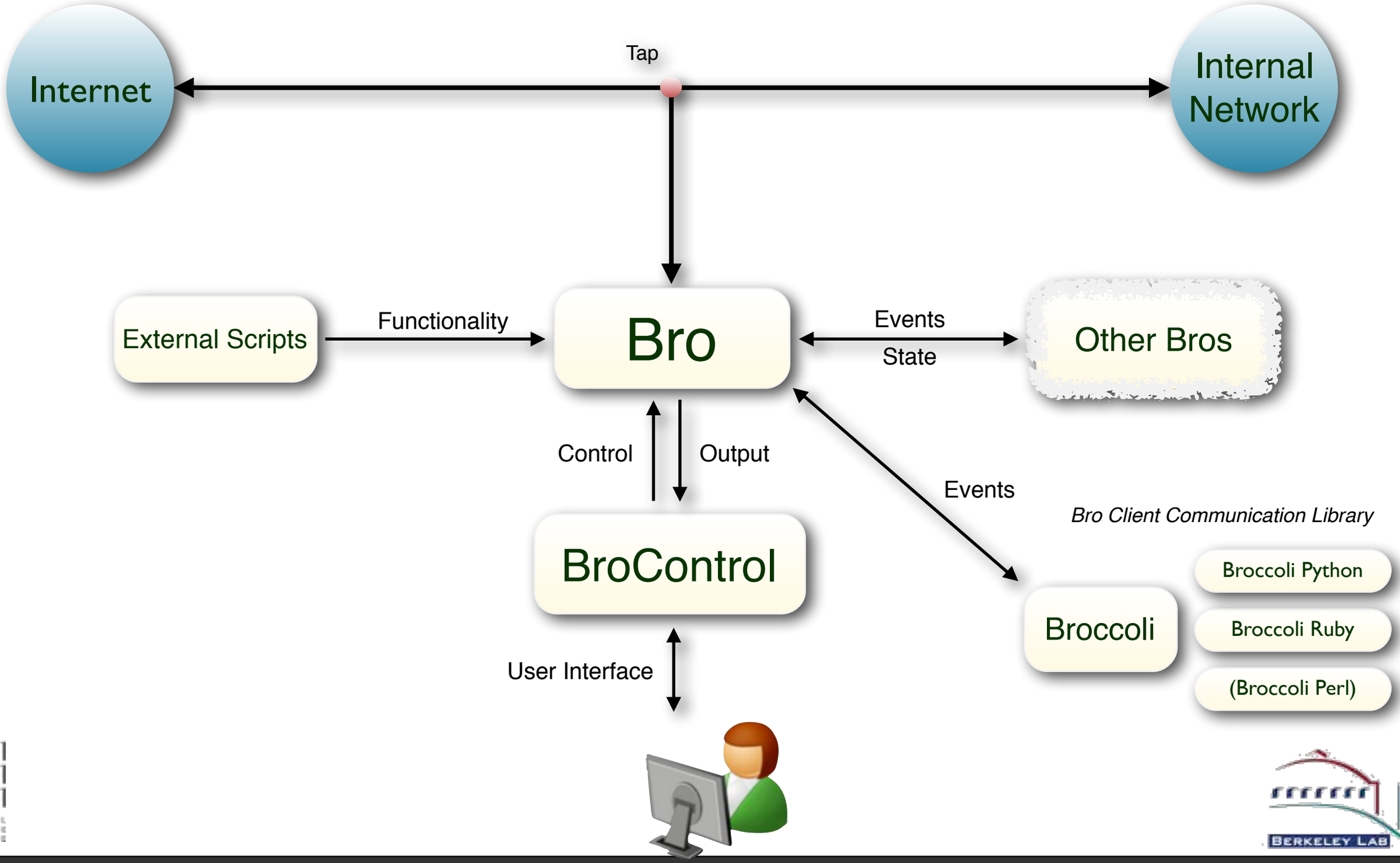
Bro Ecosystem



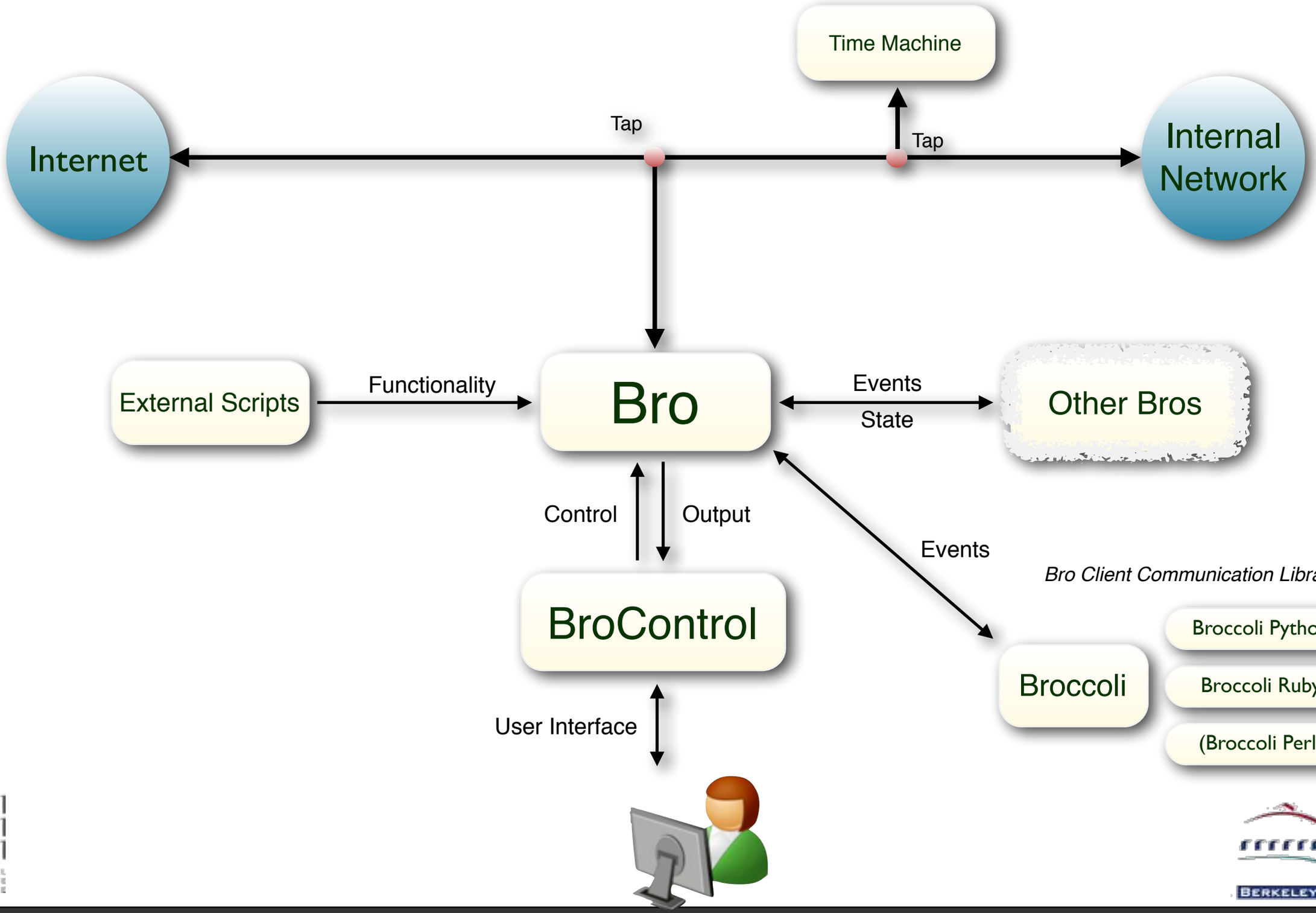
Bro Ecosystem



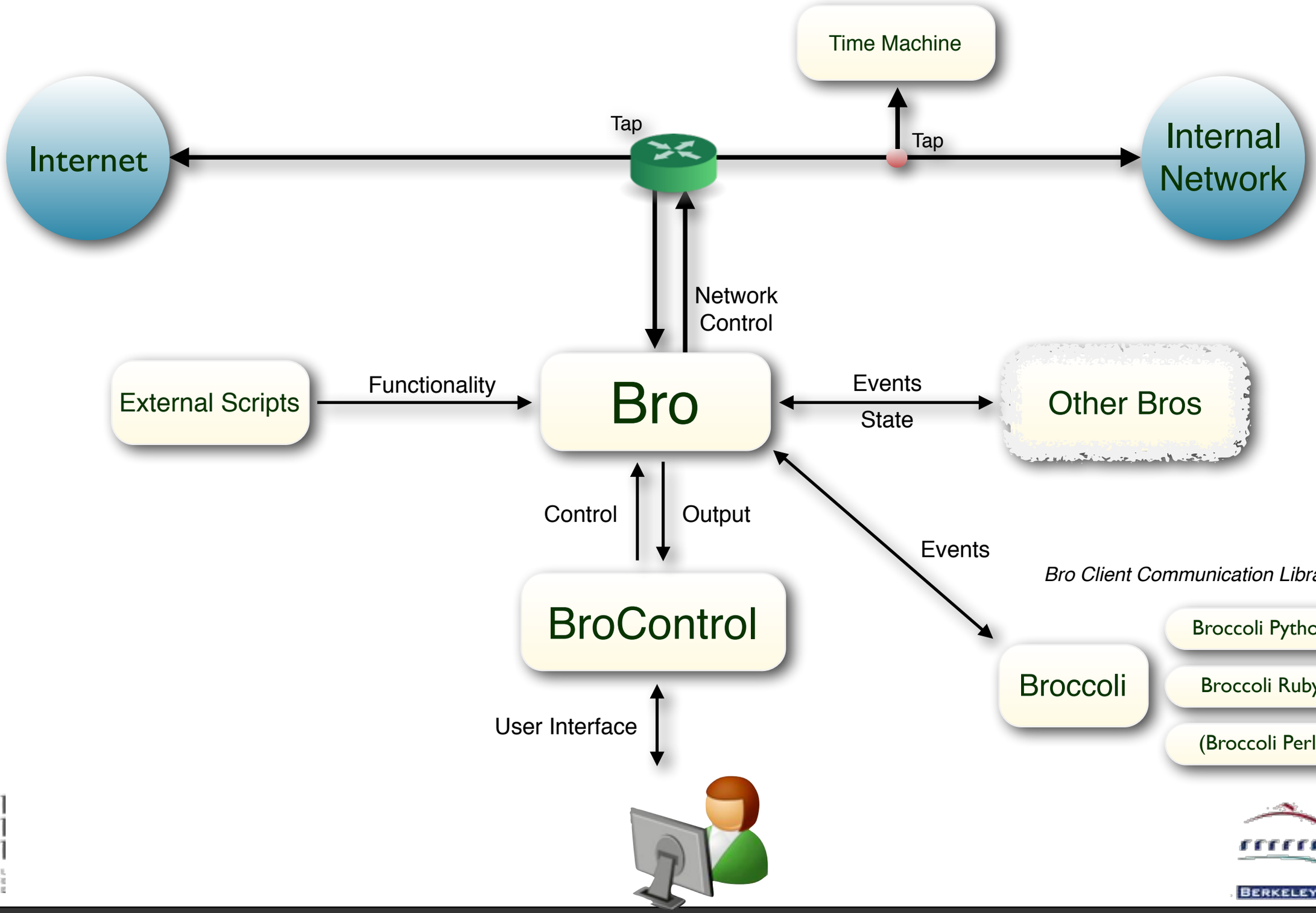
Bro Ecosystem



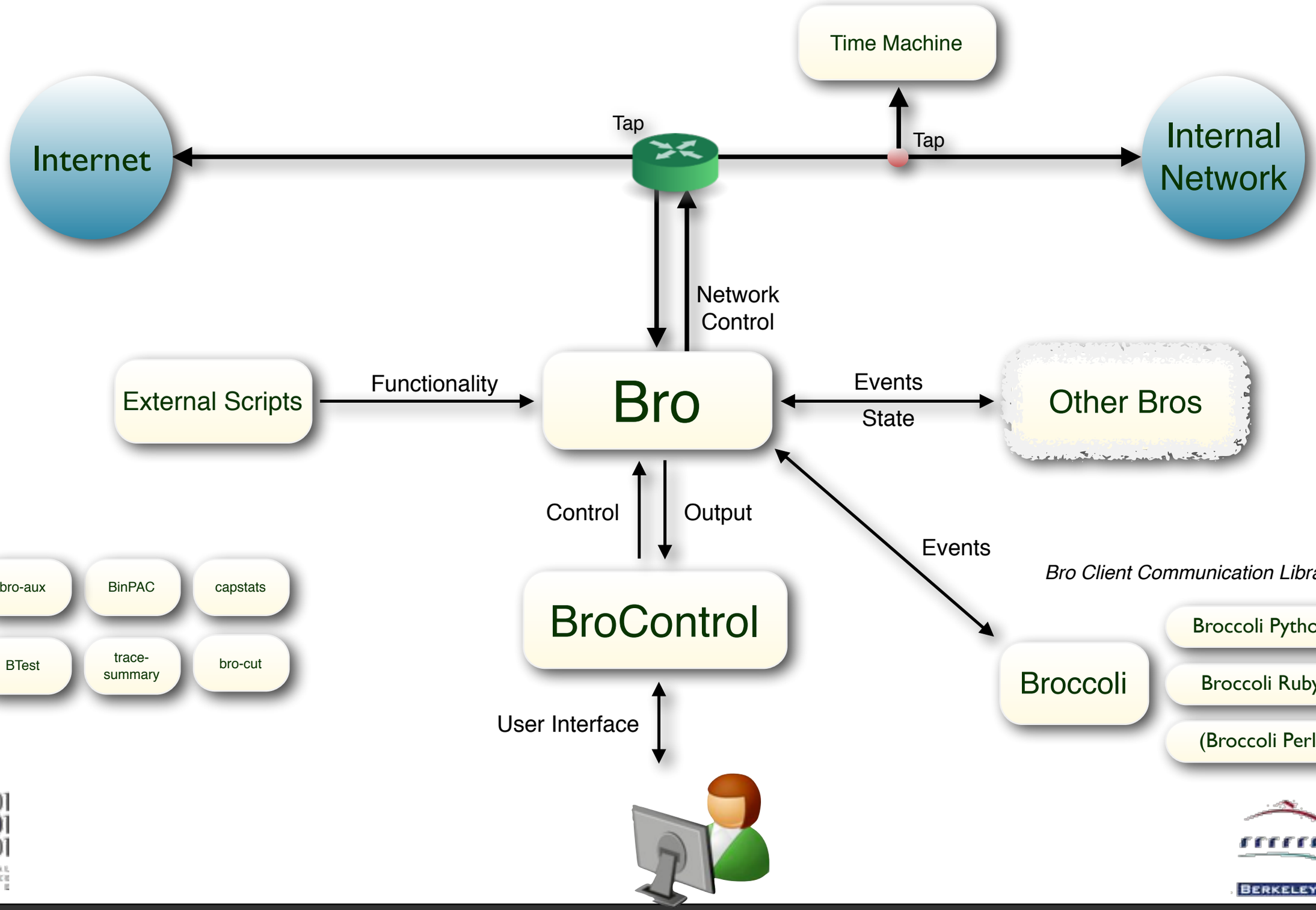
Bro Ecosystem



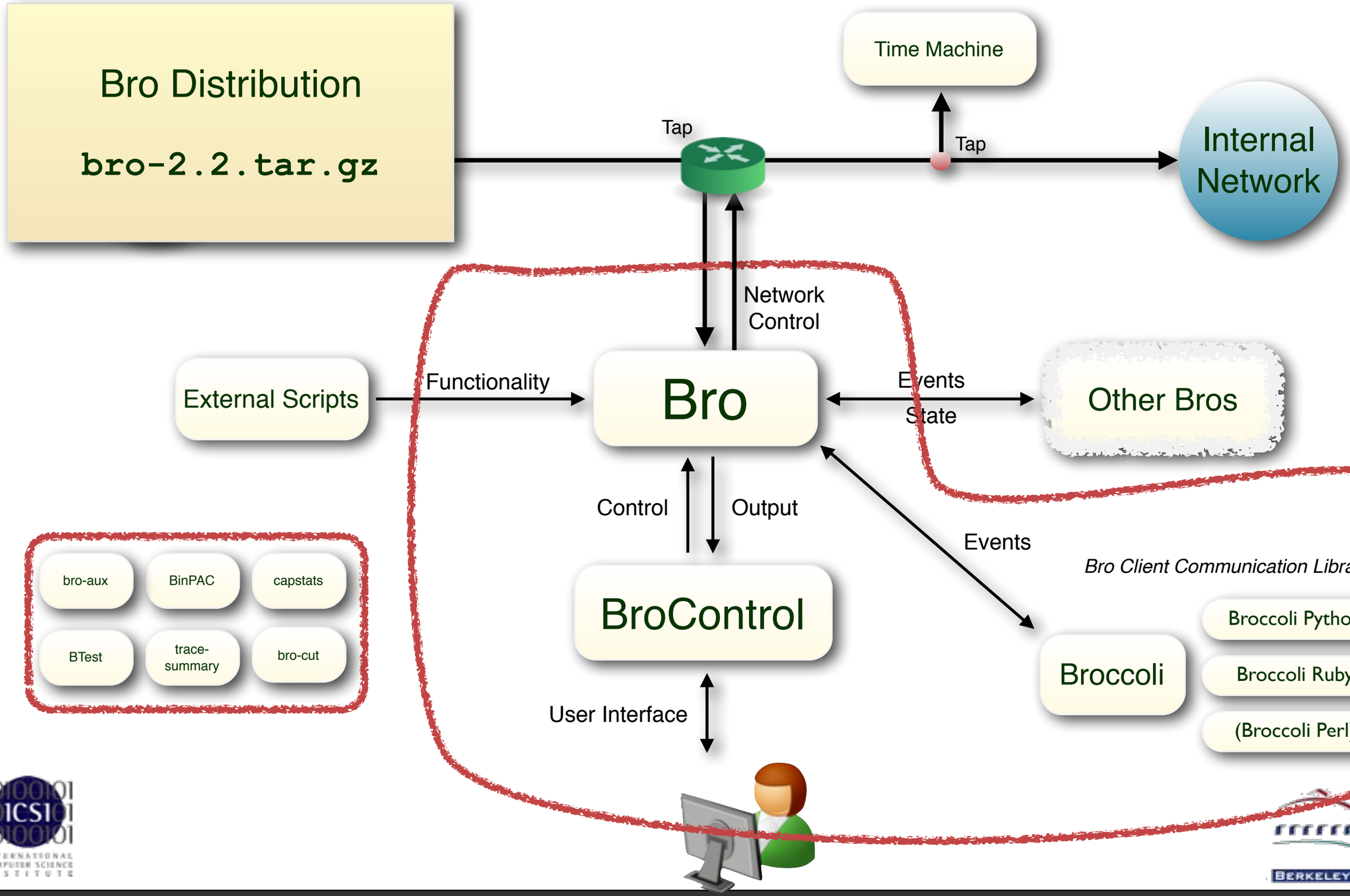
Bro Ecosystem



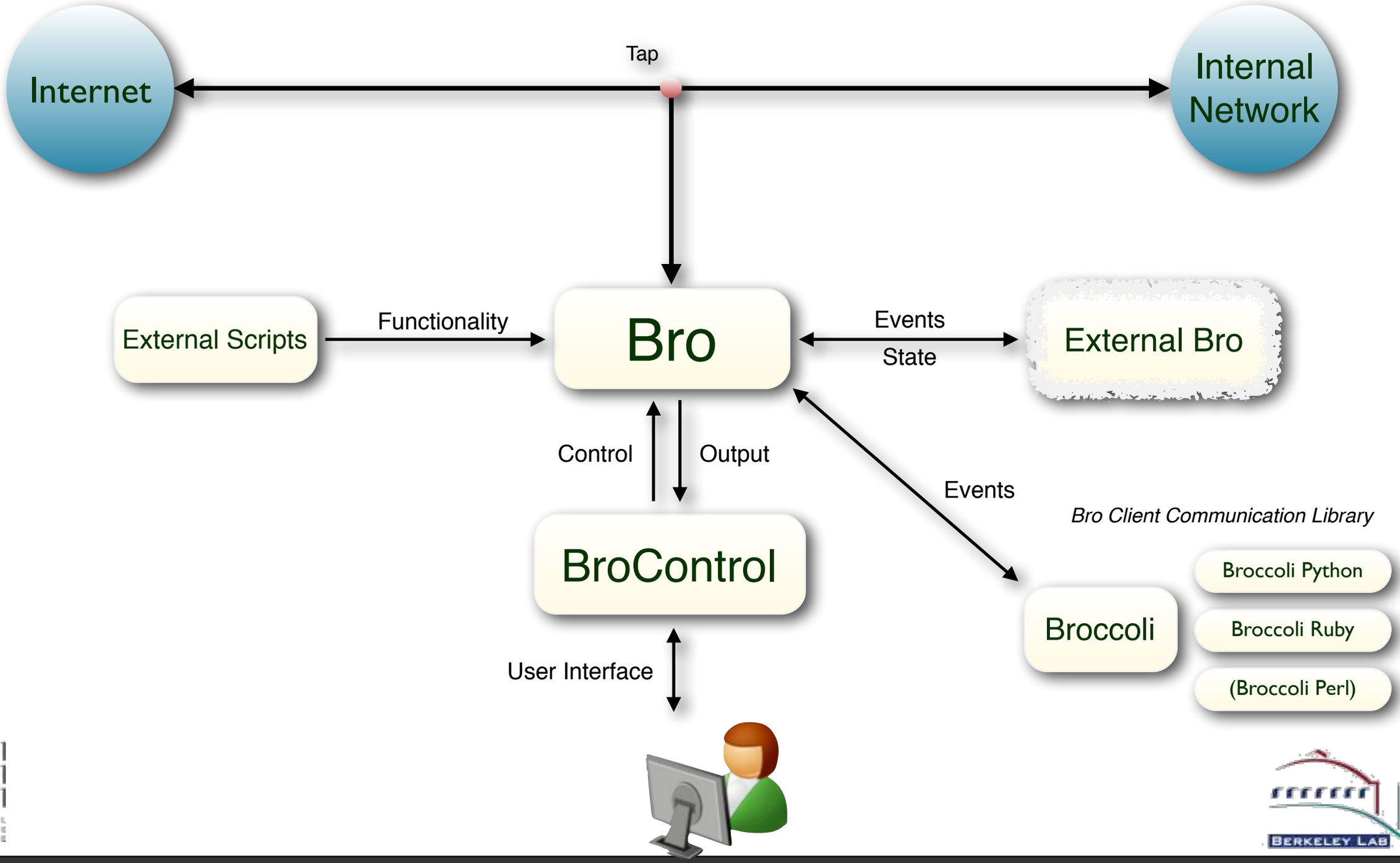
Bro Ecosystem



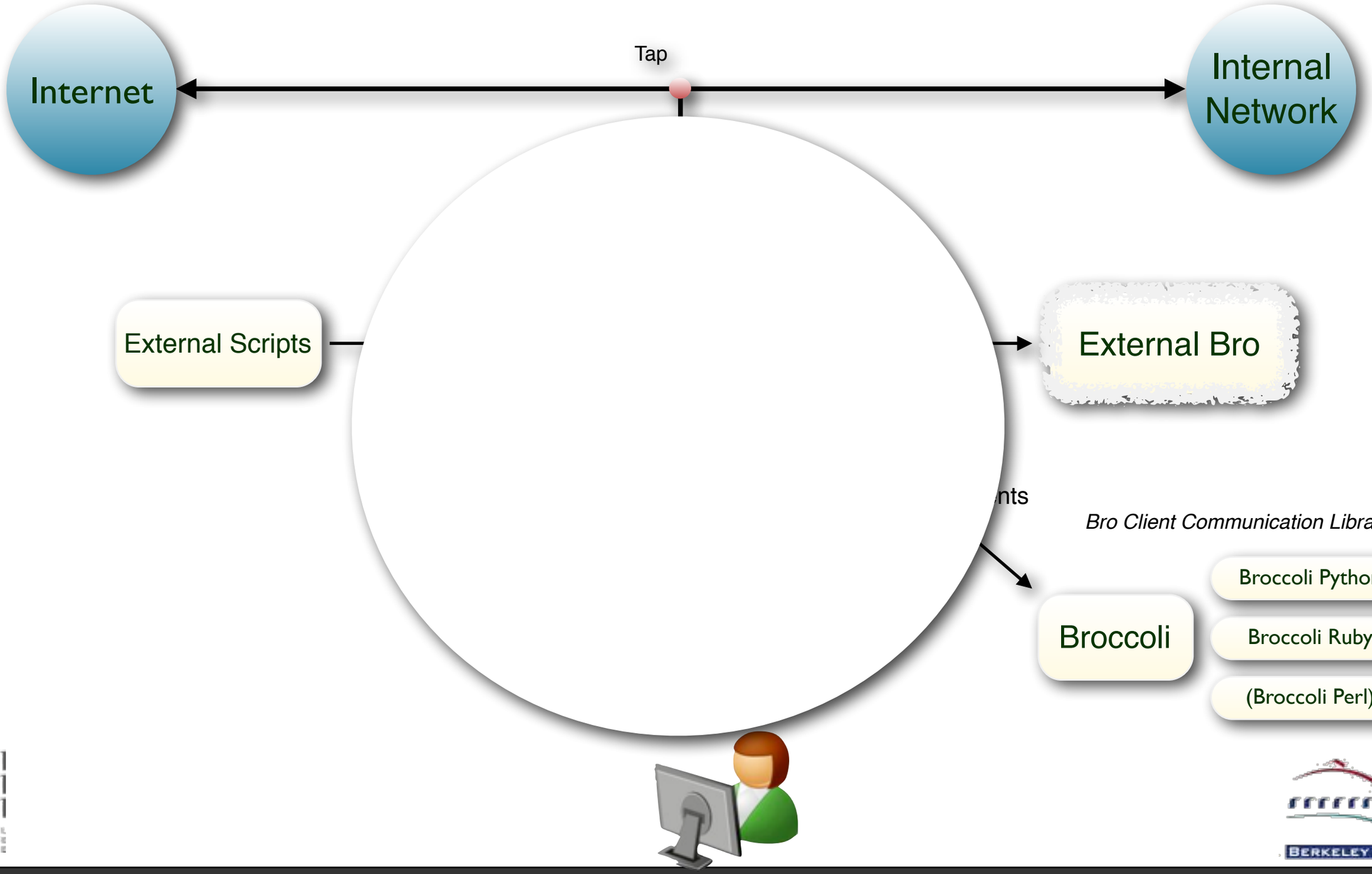
Bro Ecosystem



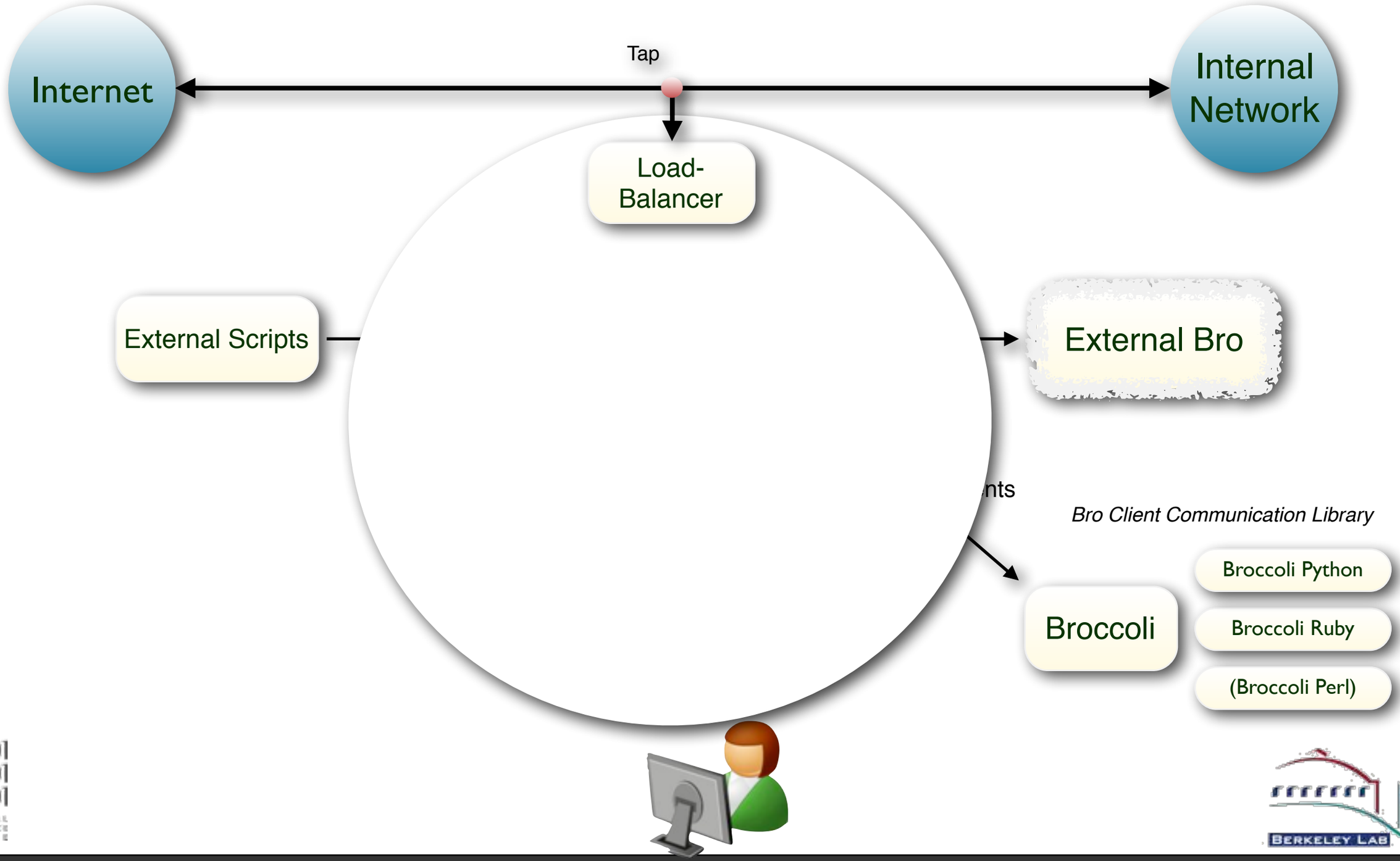
Bro *Cluster* Ecosystem



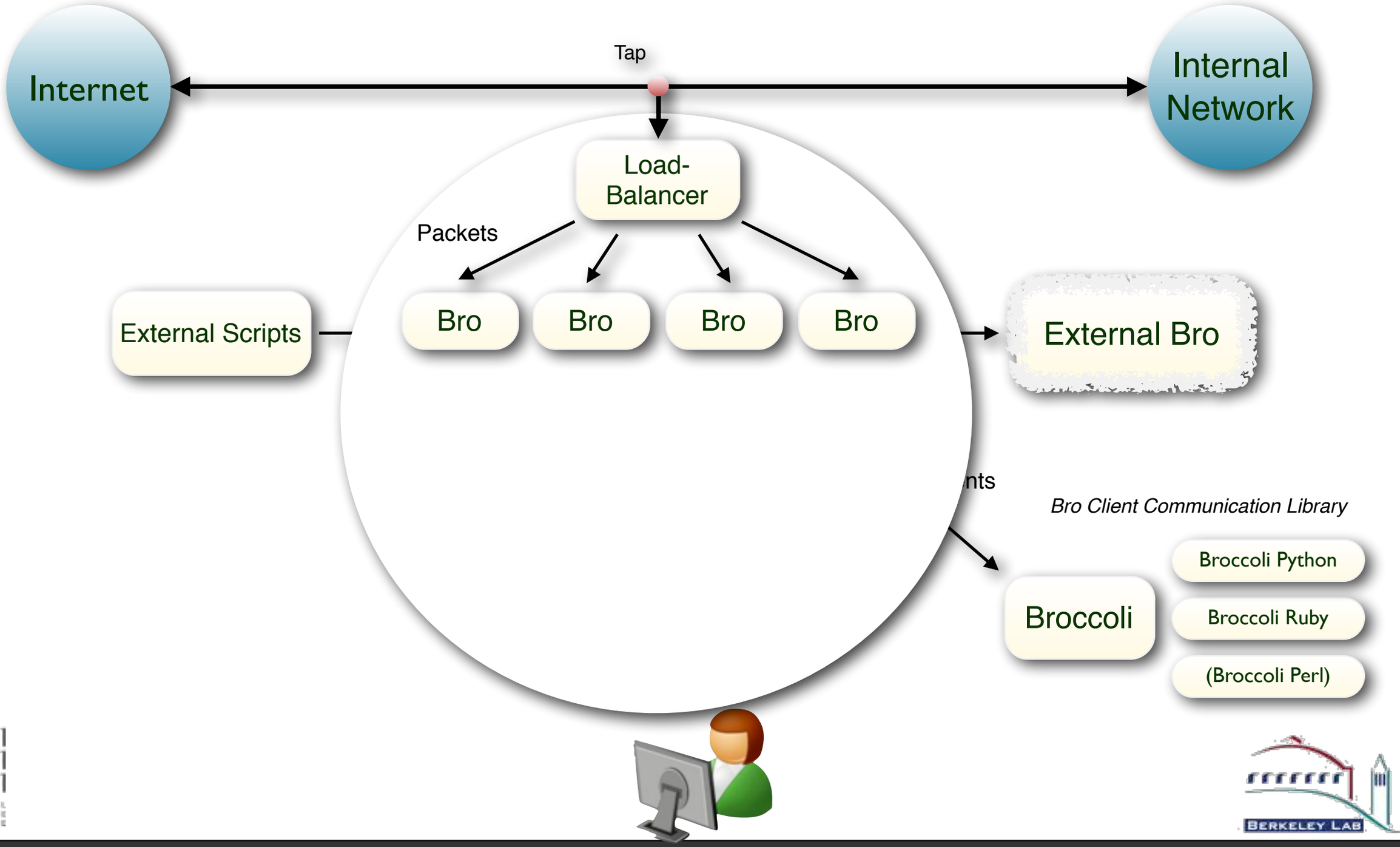
Bro *Cluster* Ecosystem



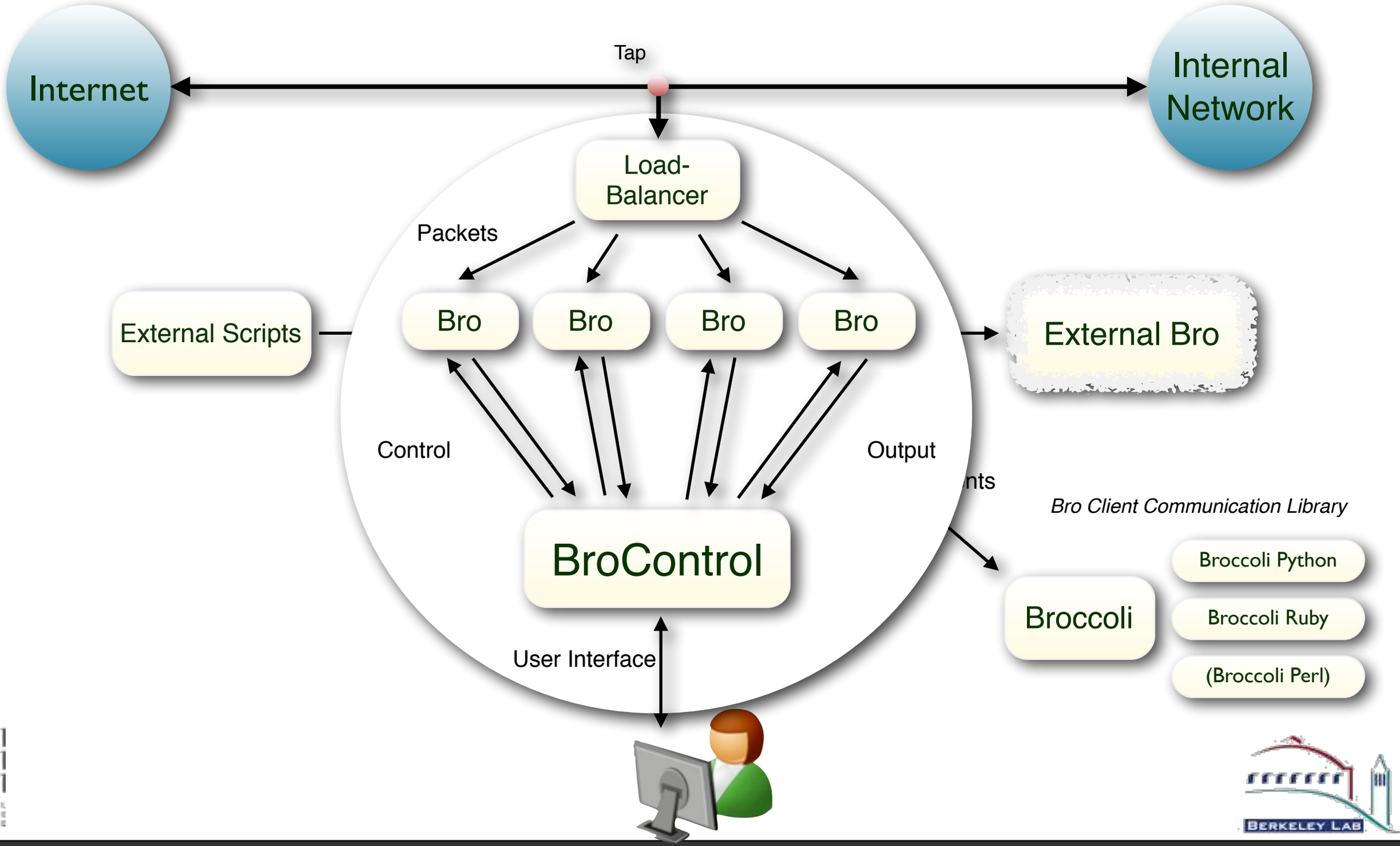
Bro *Cluster* Ecosystem



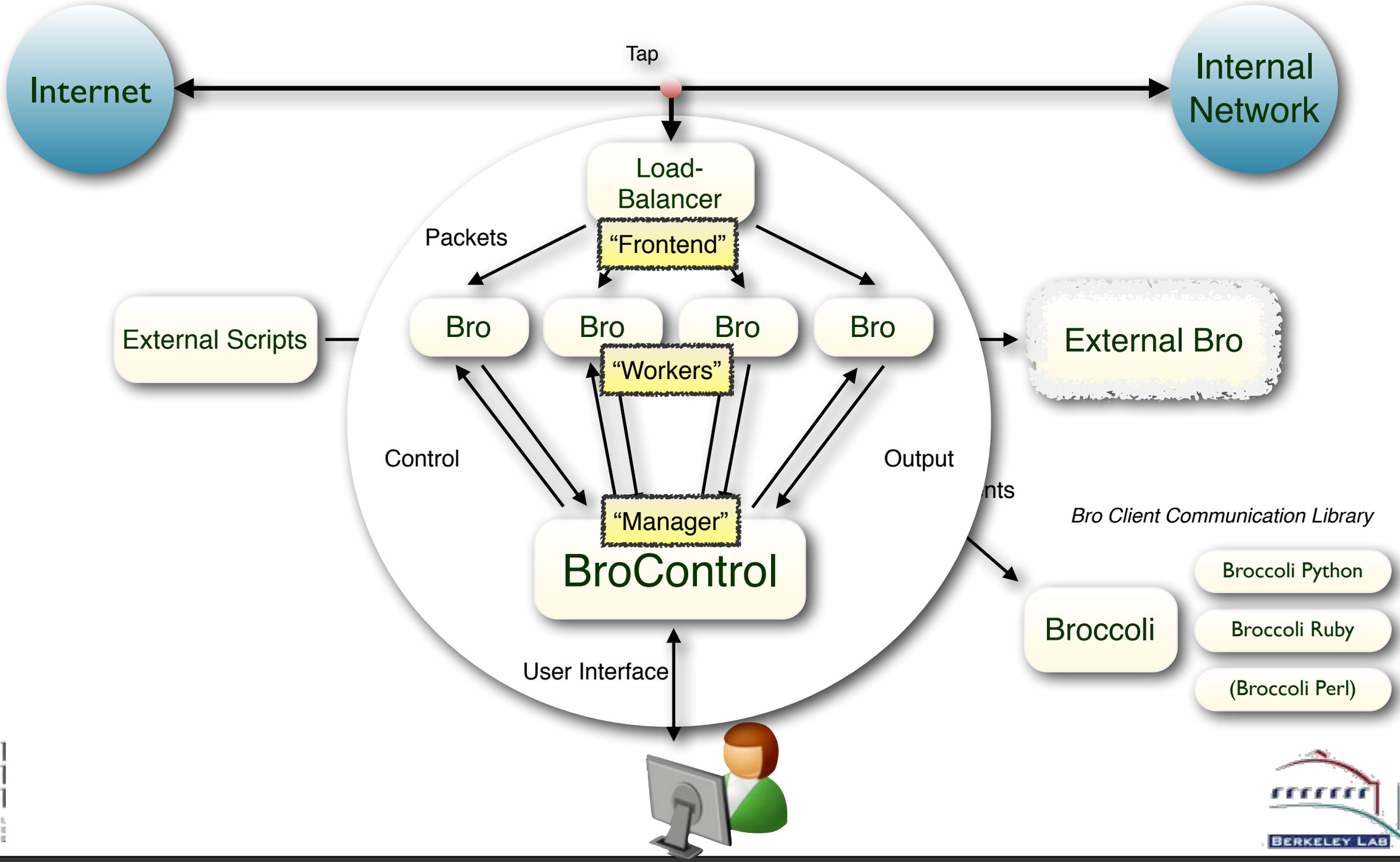
Bro *Cluster* Ecosystem



Bro *Cluster* Ecosystem



Bro *Cluster* Ecosystem



So much more …

Bro is … a Platform

Intrusion Detection	Vulnerabilit. Mgmt	File Analysis	Traffic Measure-ment	Traffic Control	Compliance Monitoring

There's much more I could talk about …

Host-level integration	More File Analysis
Data import and export	More Protocols
Automatic Reaction	More File Analysis
Monitoring Internal Networks	100Gb/s Networks
Measurements	Enterprise Protocols
SDN integration	Summary Statistics
Industrial Control Systems	Science DMZs
Embedded Devices	ICSI SSL Notary
Current Research	Cluster Deployment

The U.S. National Science Foundation has enabled much of this work.

Bro is coming out of almost two decades of academic research, along with extensive transition to practice efforts. NSF has supported much of that, and is currently funding a Bro Center of Expertise at the *International Computer Science Institute* and the *National Center for Supercomputing Applications*.

The Bro Project

www.bro.org
info@bro.org
@Bro_IDS

Commercial Support

www.broala.com
info@broala.com
@Broala_

broala

The U.S. National Science Foundation has enabled much of this work.

Bro is coming out of almost two decades of academic research, along with extensive transition to practice efforts. NSF has supported much of that, and is currently funding a Bro Center of Expertise at the *International Computer Science Institute* and the *National Center for Supercomputing Applications*.

The Bro Project

www.bro.org
info@bro.org
@Bro_IDS

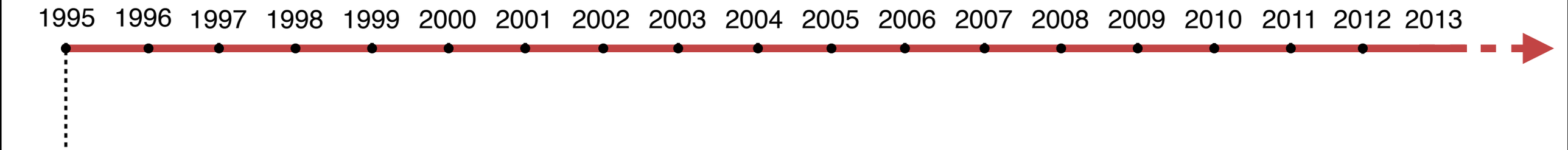
Commercial Support

www.broala.com
info@broala.com
@Broala_

broala

The End

Bro History


1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
Vern writes 1st line of code
Bro Center

Bro History

1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013

Vern writes 1st line of code

LBNL starts using Bro operationally

v0.2
1st CHANGES entry

v0.4
HTTP analysis
Scan detector
IP fragments
Linux support

v0.6
RegExps
Login analysis

v0.7a48
Consistent CHANGES

v0.7a90
Profiling
State Mgmt

v0.7a175/0.8aX
Signatures
SMTP
IPv6 support
User manual

0.8a37
Communication
Persistence
Namespaces
Log Rotation

v0.8aX/0.9aX
SSL/SMB
STABLE releases
BroLite

v1.0
BinPAC
IRC/RPC analyzers
64-bit support
Sane version numbers

v1.1/v1.2
when Stmt
Resource tuning
Broccoli
DPD

v1.3
Ctor expressions
GeoIP
Conn Compressor

v1.4
DHCP/BitTorrent
HTTP entities
NetFlow
Bro Lite Deprecated

v1.5
BroControl

Bro SDCI

v2.0
New Scripts

v2.1
IPv6
Input Framew.

v2.2
File Analysis
Summary Stat.

Bro Center

Bro History

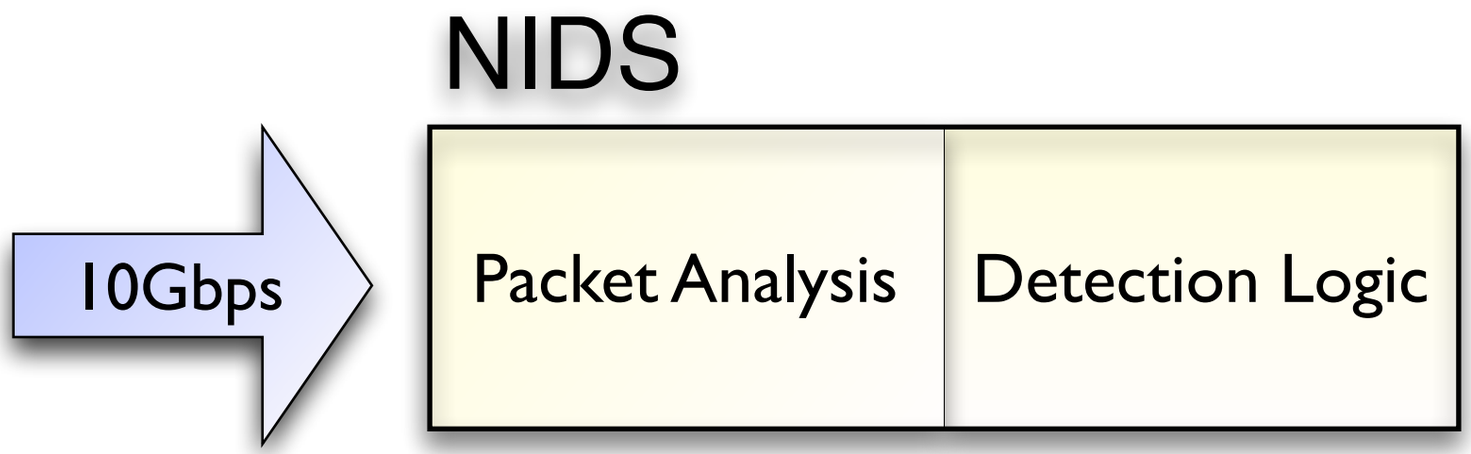
Academic Publications

1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013

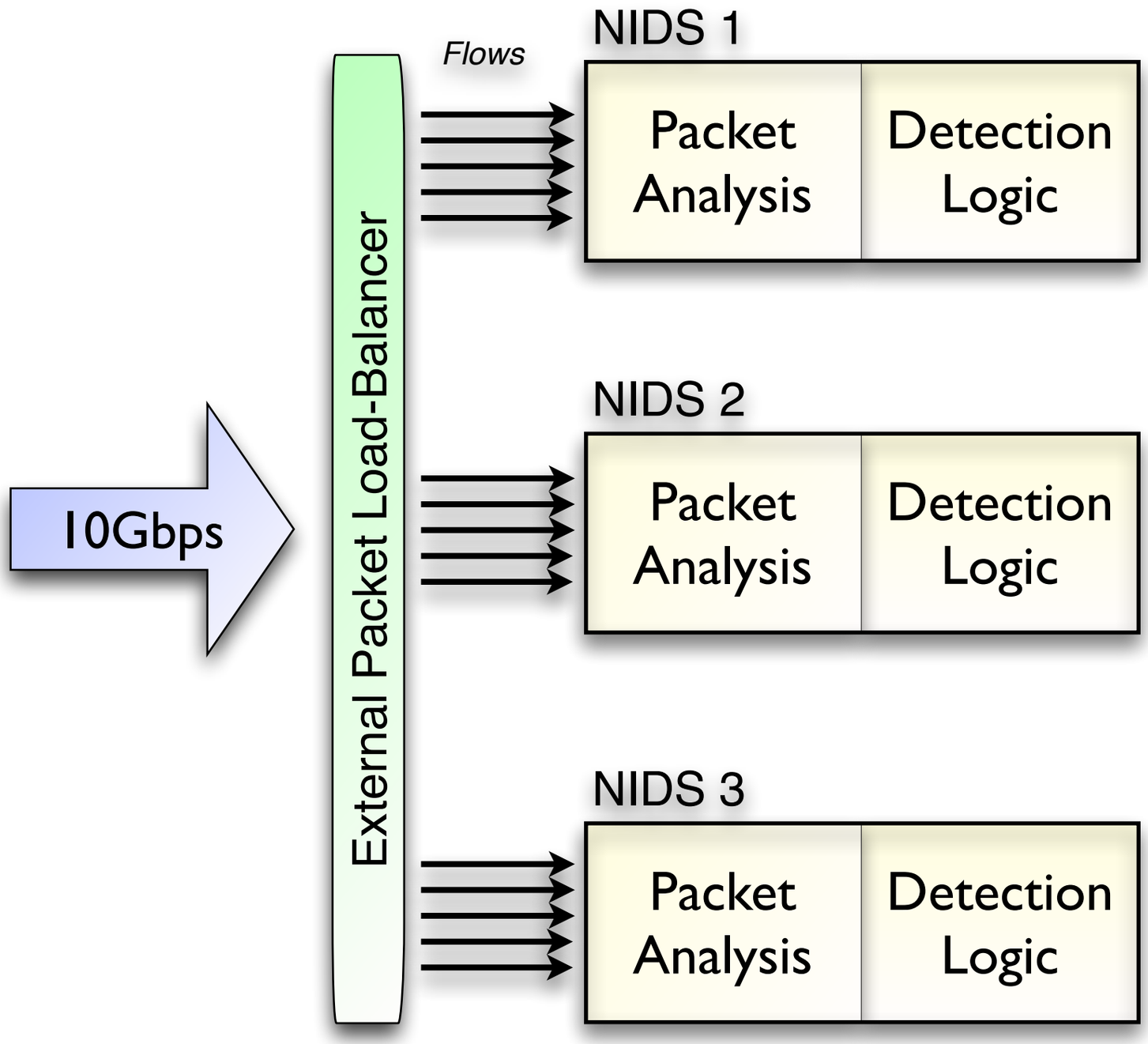
- 1995 — Vern writes 1st line of code
- 1996 — LBNL starts using Bro operationally
- 1997 — v0.2 1st CHANGES entry
- 1998 — USENIX Paper; v0.4 HTTP analysis, Scan detector, IP fragments, Linux support
- 1999 — v0.6 RegExps, Login analysis
- 2000 — Stepping Stone Detector; v0.7a48 Consistent CHANGES
- 2001 — v0.7a90 Profiling, State Mgmt
- 2002 — v0.7a175/0.8aX Signatures, SMTP, IPv6 support, User manual
- 2003 — Anonymizer, Active Mapping, Context Signat.; 0.8a37 Communication, Persistence, Namespaces, Log Rotation
- 2004 — TRW, State Mgmt., Independ. State; v0.8aX/0.9aX SSL/SMB, STABLE releases, BroLite
- 2005 — Host Context, Time Machine, Enterprise Traffic; v1.0 BinPAC, IRC/RPC analyzers, 64-bit support, Sane version numbers
- 2006 — BinPAC, DPD, 2nd Path; v1.1/v1.2 *when* Stmt, Resource tuning, Broccoli, DPD
- 2007 — Bro Cluster, Shunt; v1.3 Ctor expressions, GeoIP, Conn Compressor
- 2008 — Autotuning; v1.4 DHCP/BitTorrent, HTTP entities, NetFlow, Bro Lite Deprecated
- 2009 — Parallel Prototype; v1.5 BroControl
- 2010 — **Bro SDCI** (NSF); v2.0 New Scripts
- 2011 — v2.1 IPv6, Input Framew.
- 2012 — Input Framework
- 2013 — v2.2 File Analysis, Summary Stat.; **Bro Center** (NSF); broala

Load-balancing Architecture

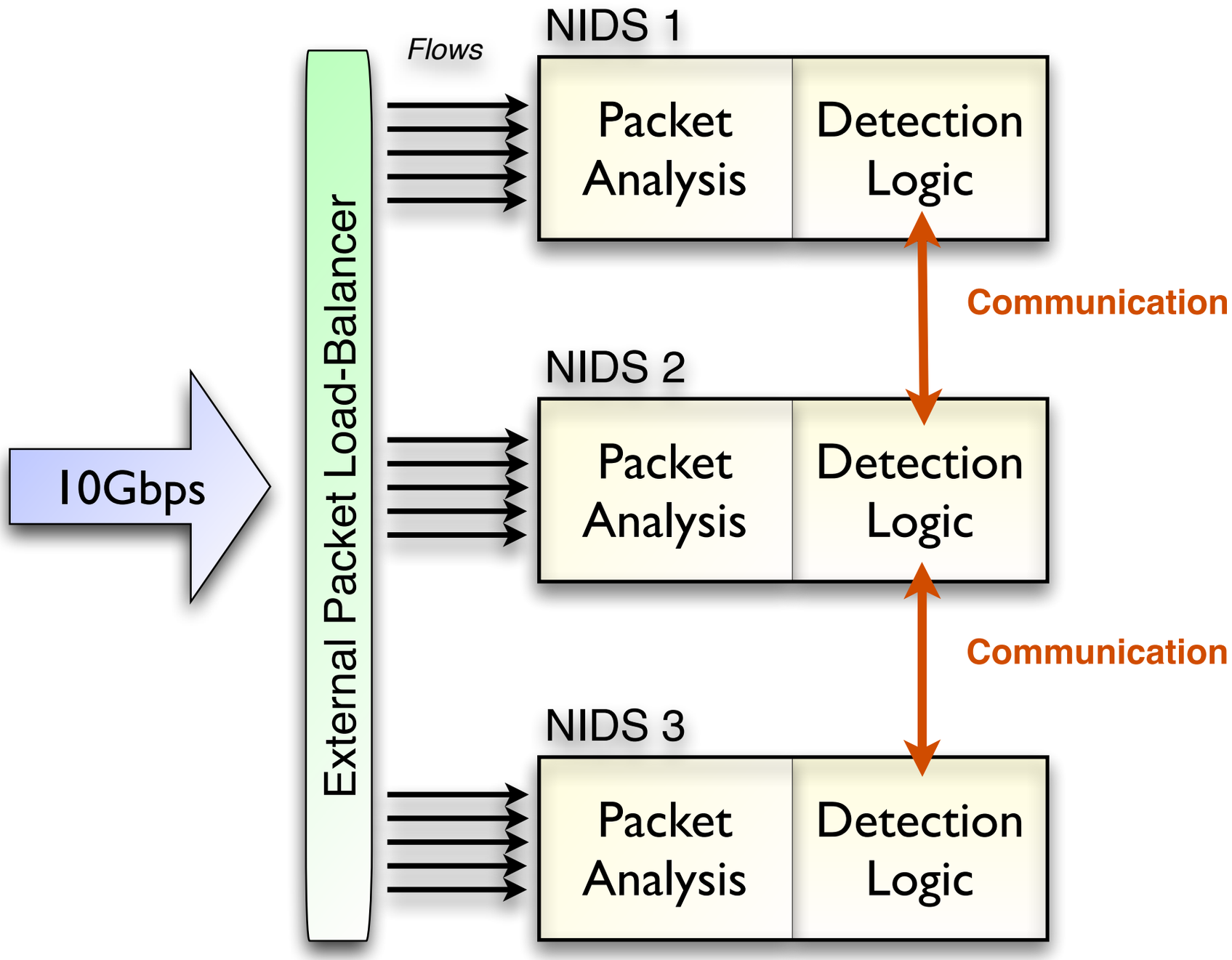
Load-balancing Architecture



Load-balancing Architecture

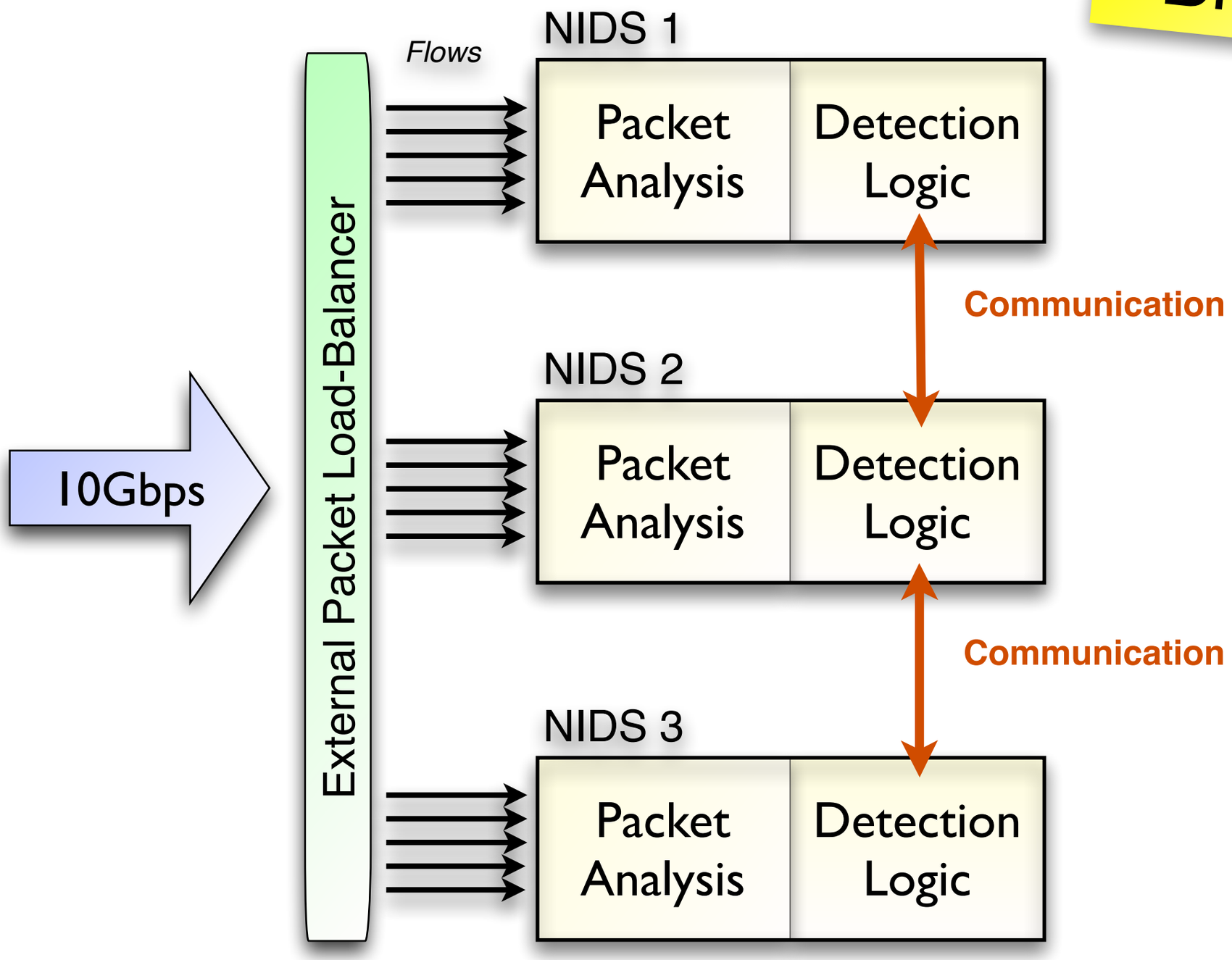


Load-balancing Architecture

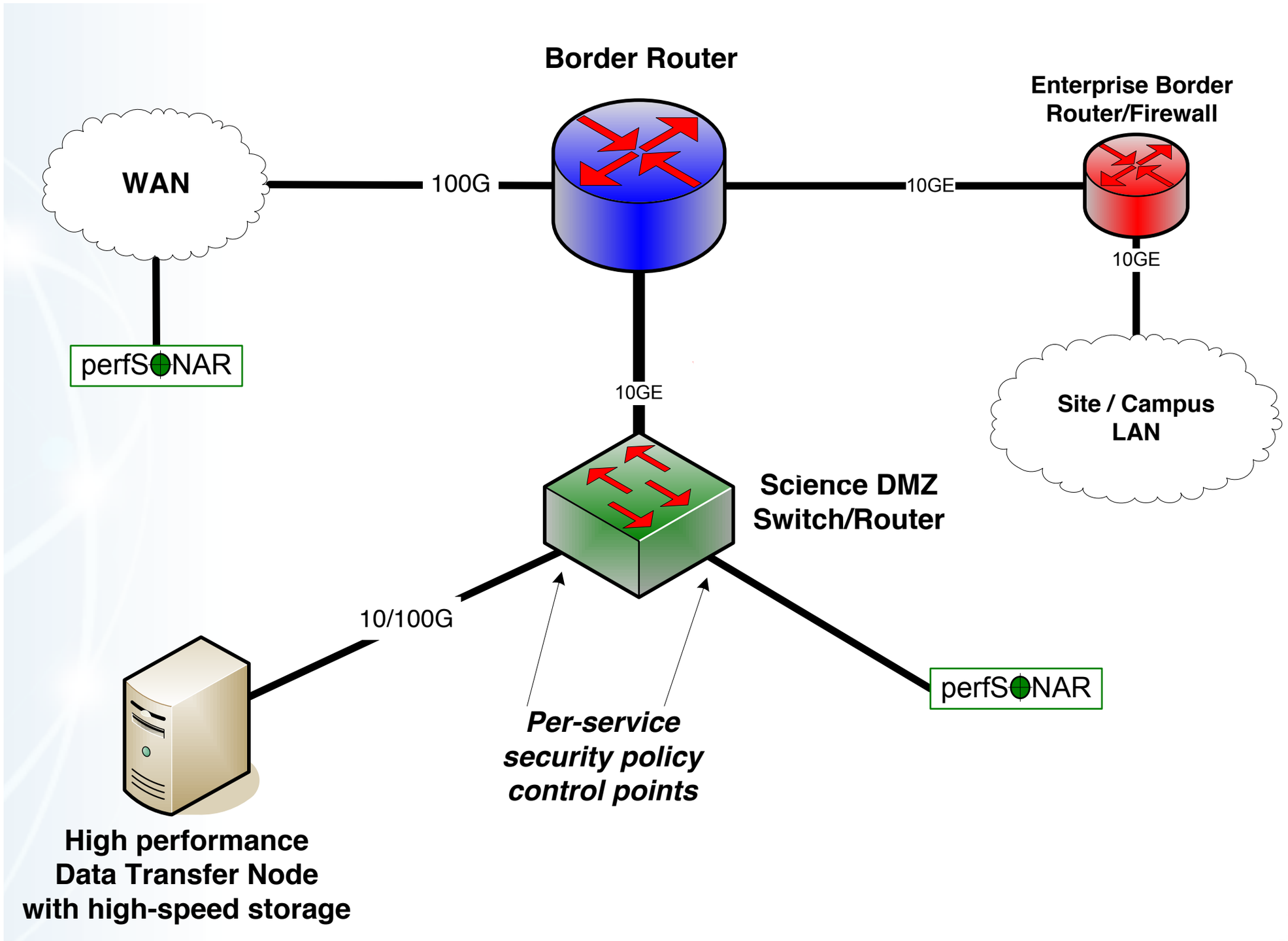


Load-balancing Architecture

"Bro Cluster"

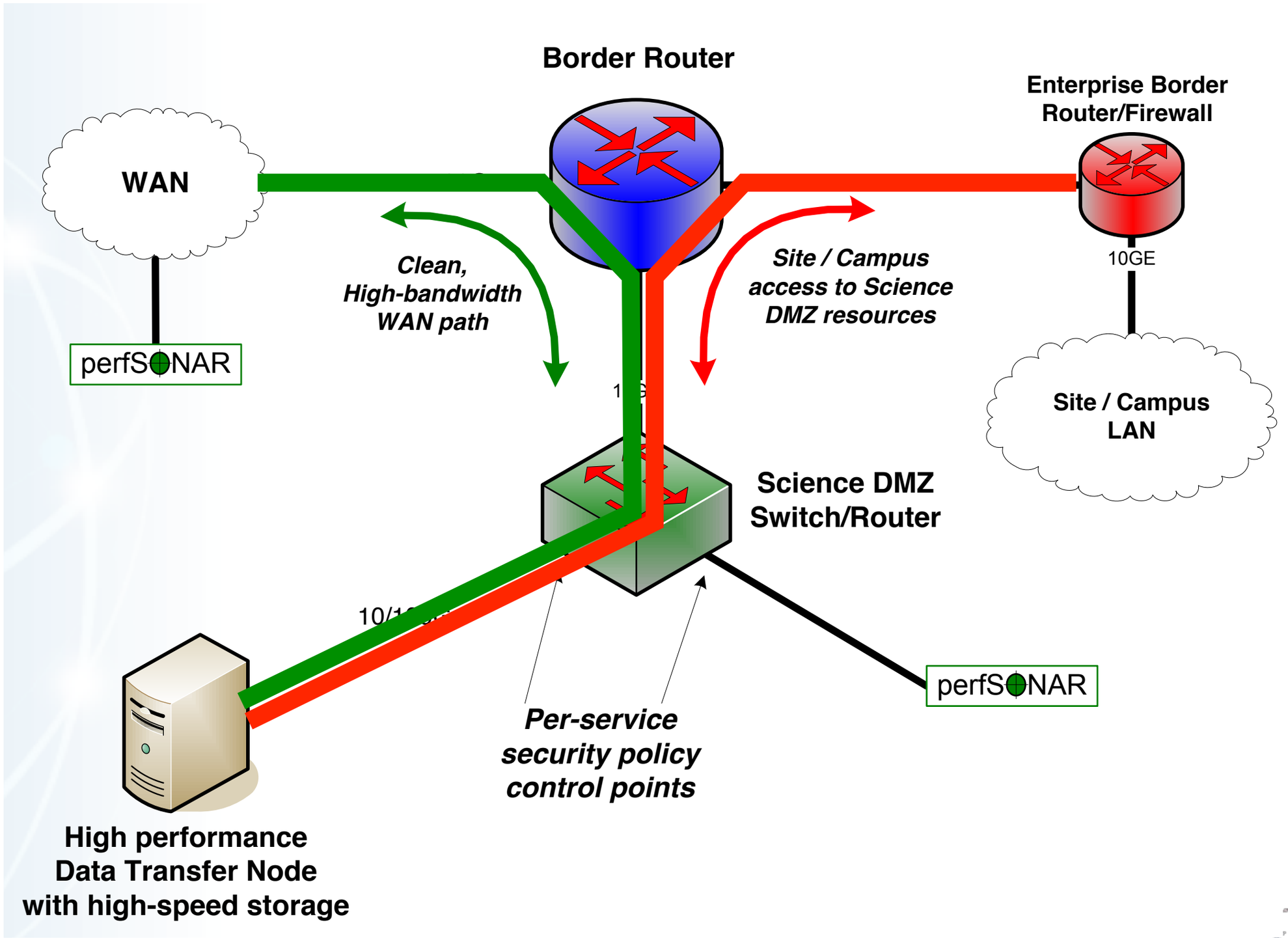


Science DMZs



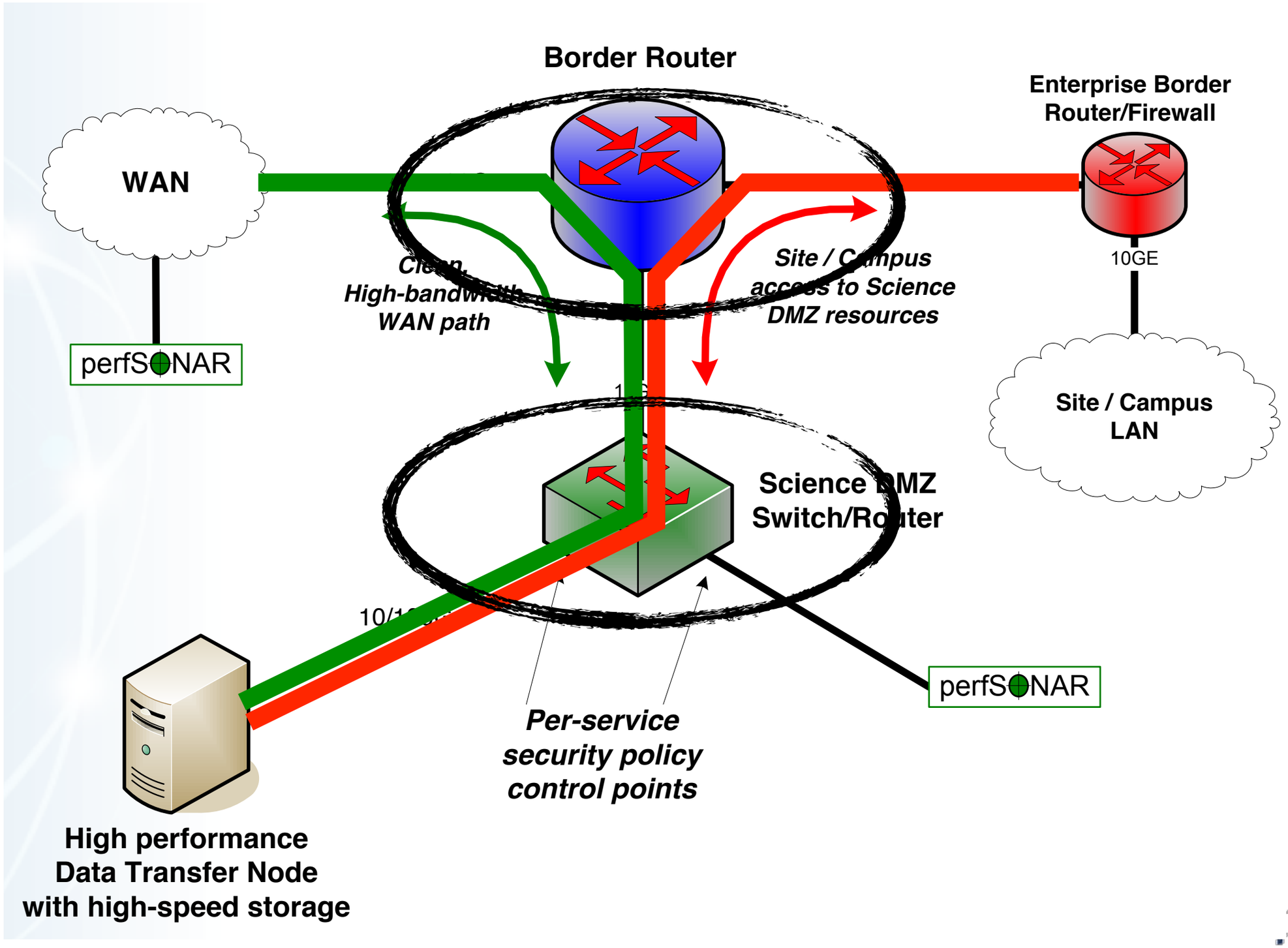
Source: ESNet

Science DMZs



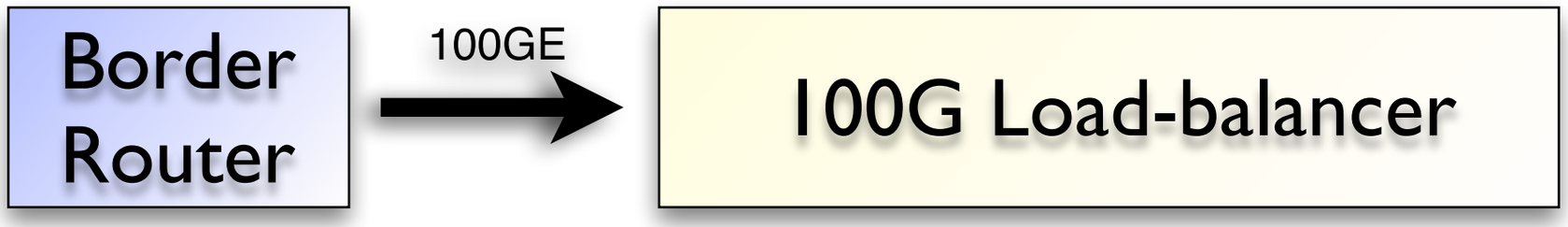
Source: ESNet

Science DMZs

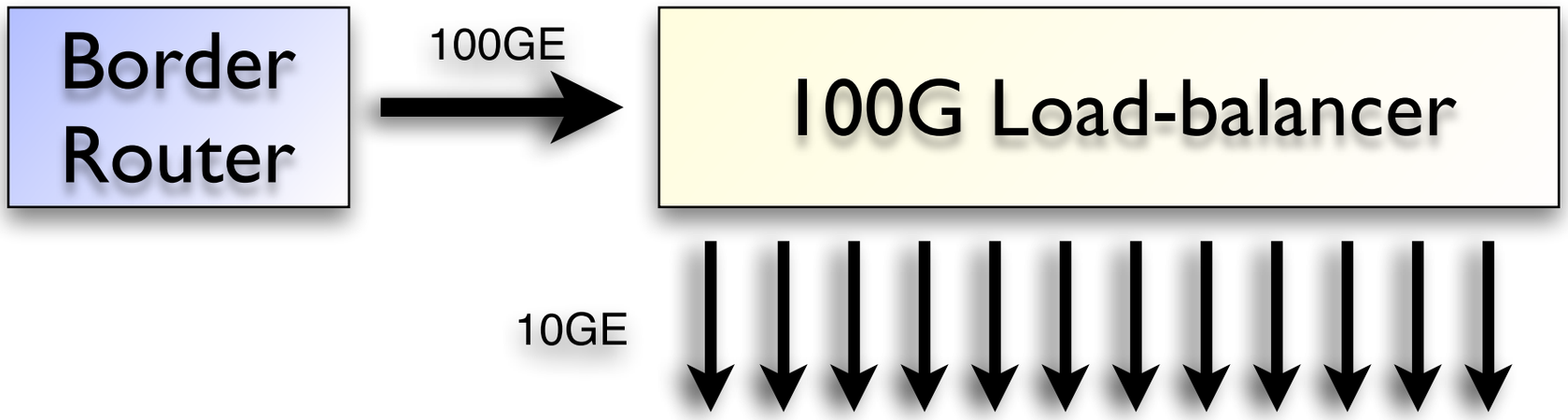


Source: ESNet

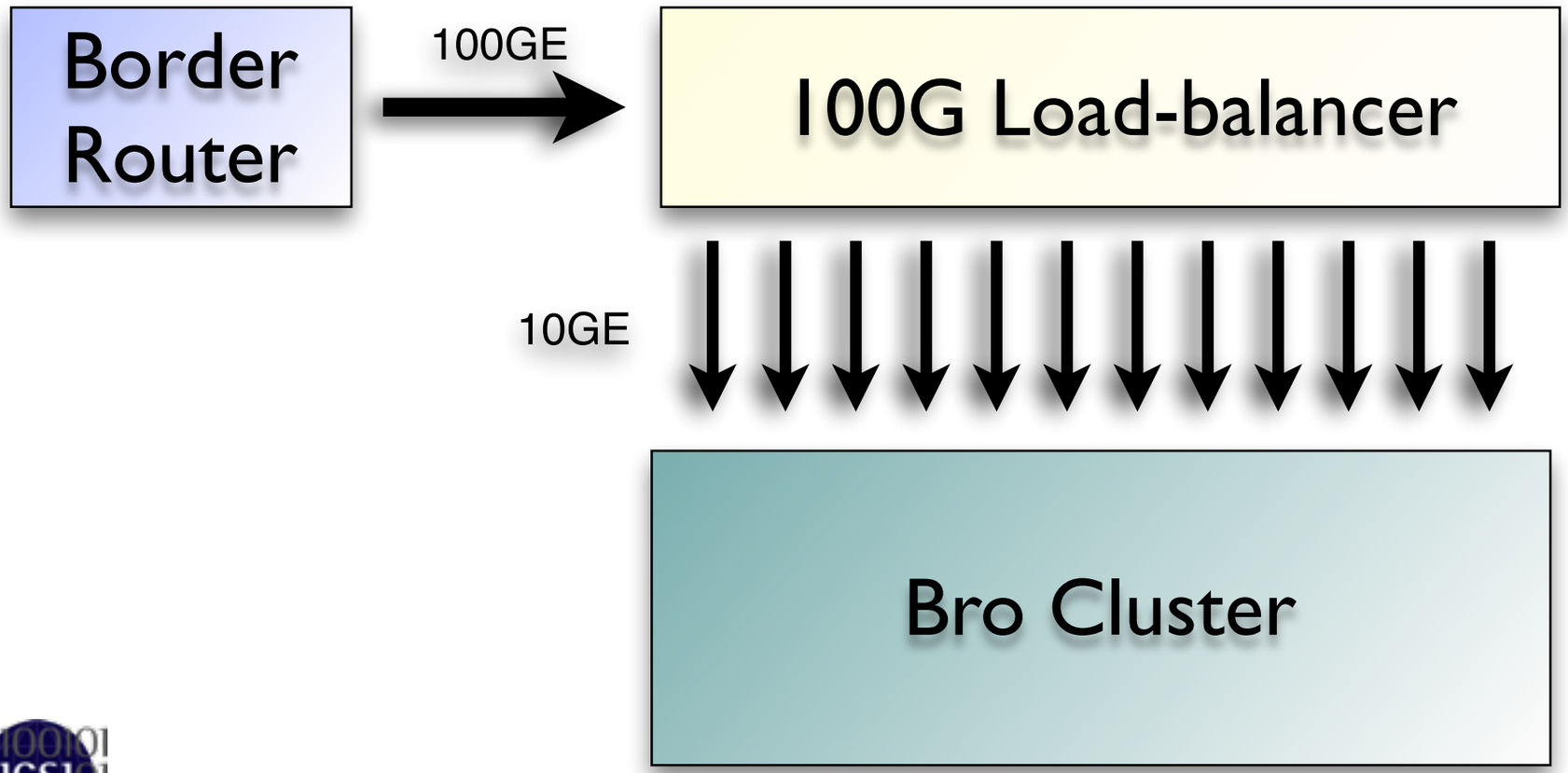
100 Gb/s Cluster



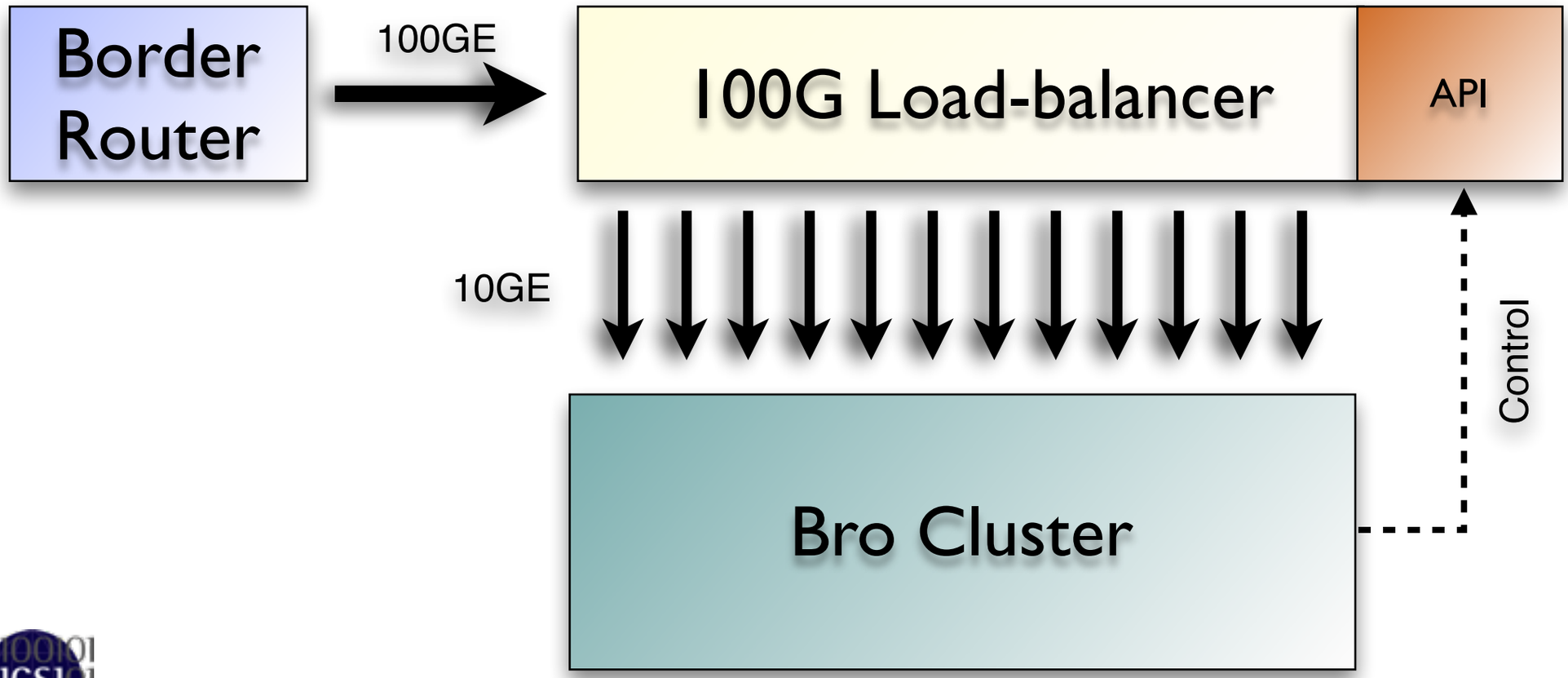
100 Gb/s Cluster



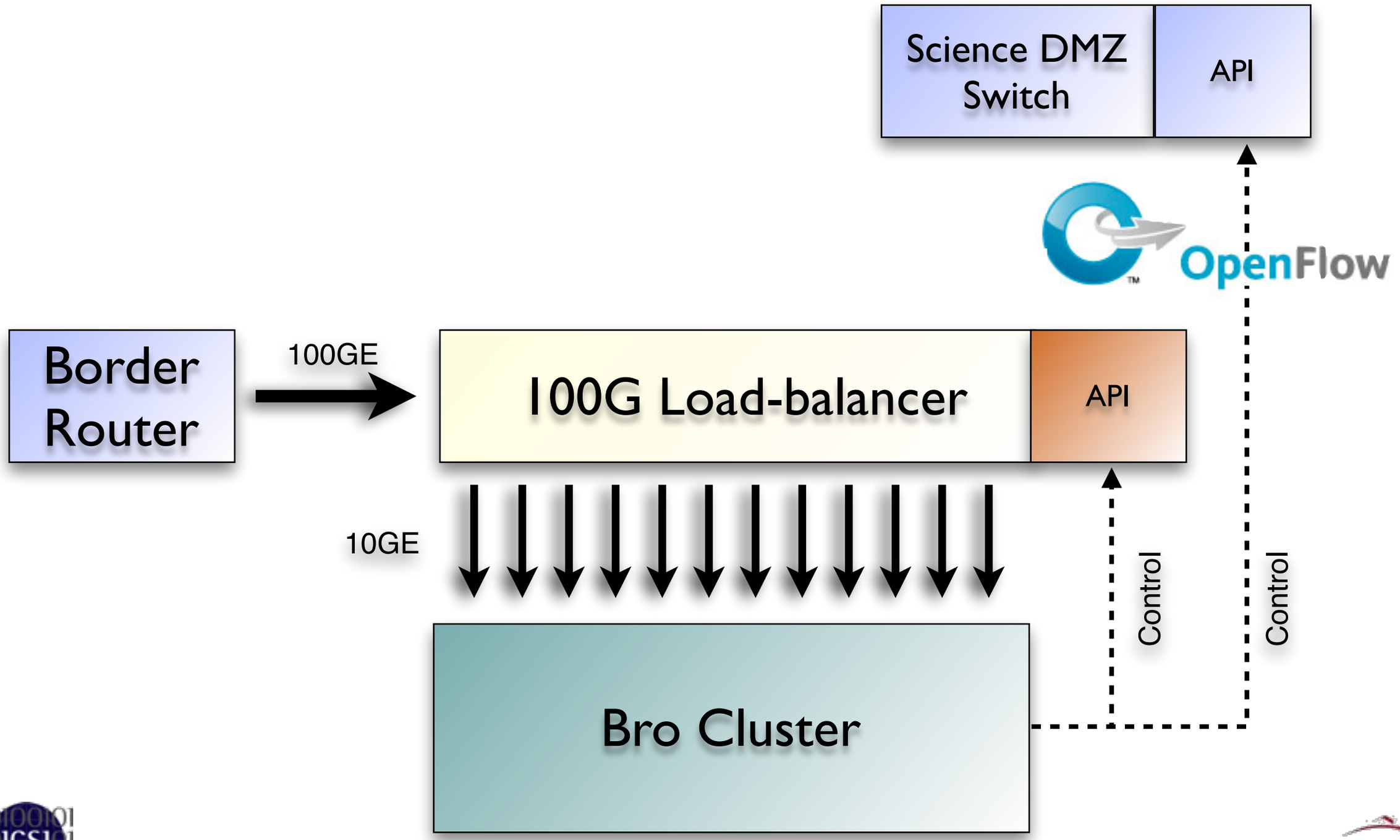
100 Gb/s Cluster



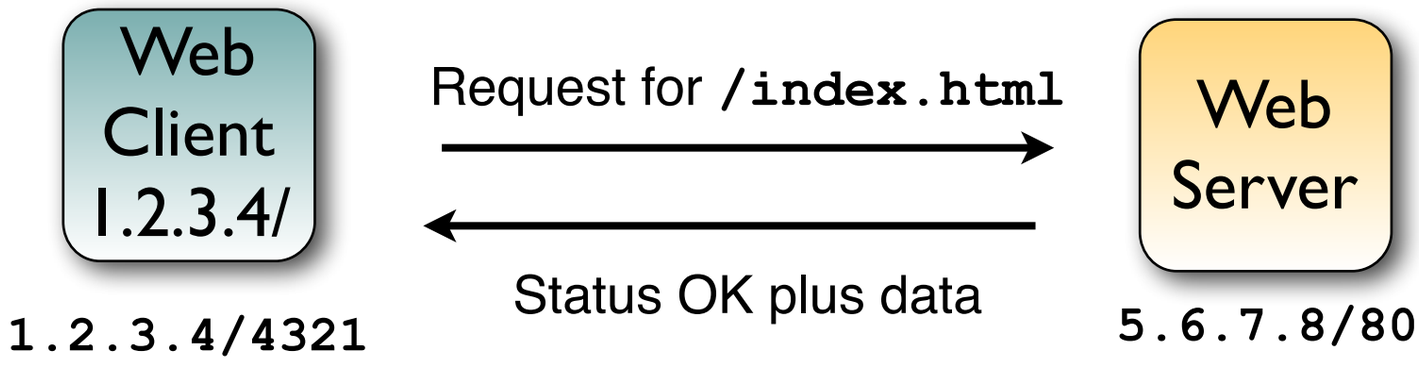
100 Gb/s Cluster



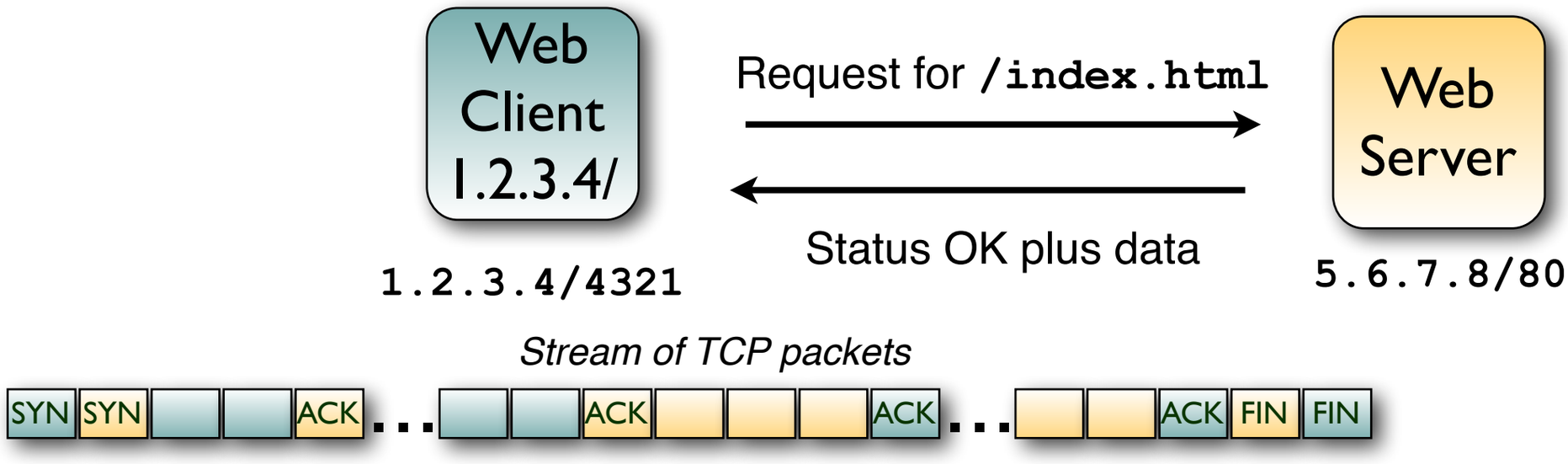
100 Gb/s Cluster



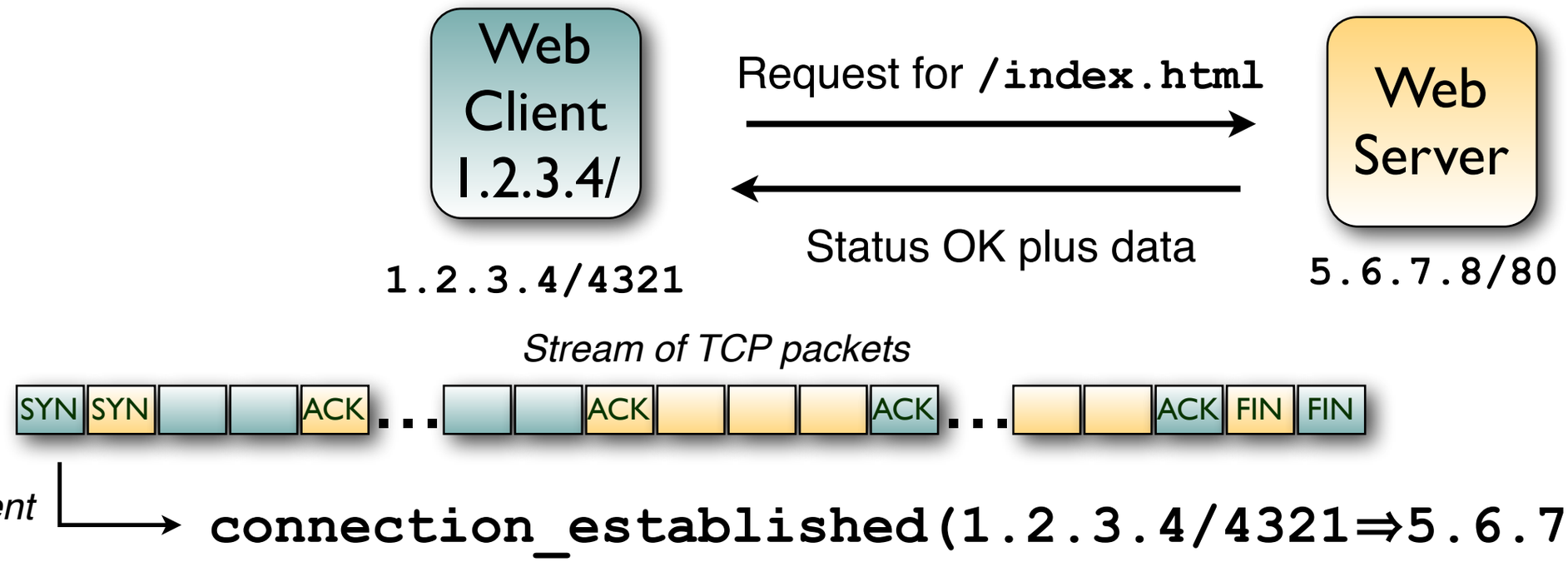
Event Model


Web Client 1.2.3.4/
1.2.3.4/4321
Request for /index.html
Status OK plus data
Web Server
5.6.7.8/80

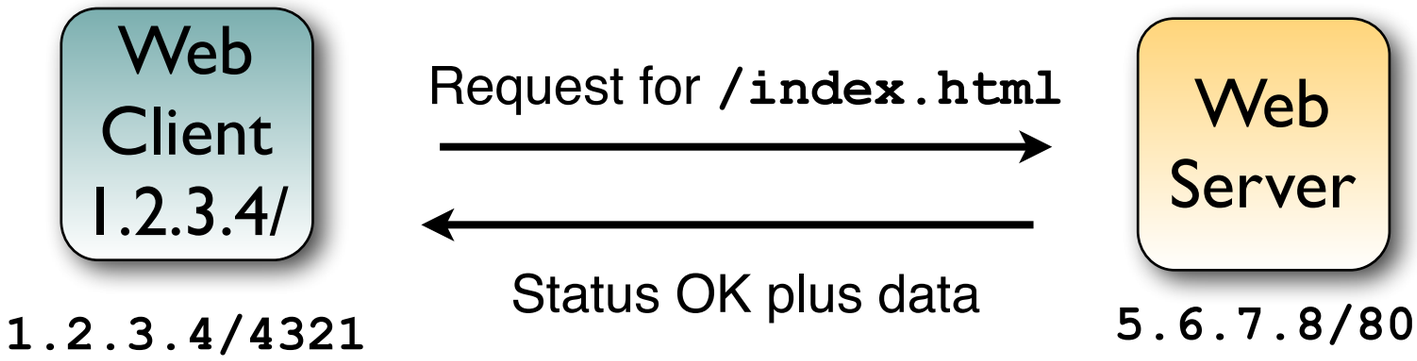
Event Model


Web Client 1.2.3.4/
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80
Stream of TCP packets
SYN
SYN
ACK
ACK
ACK
ACK
FIN
FIN

Event Model


Web Client 1.2.3.4/
Request for /index.html
Web Server
Status OK plus data
1.2.3.4/4321
5.6.7.8/80
Stream of TCP packets
SYN
SYN
ACK
ACK
ACK
ACK
FIN
FIN
Event
connection_established(1.2.3.4/4321⇒5.6.7.8/80)

Event Model



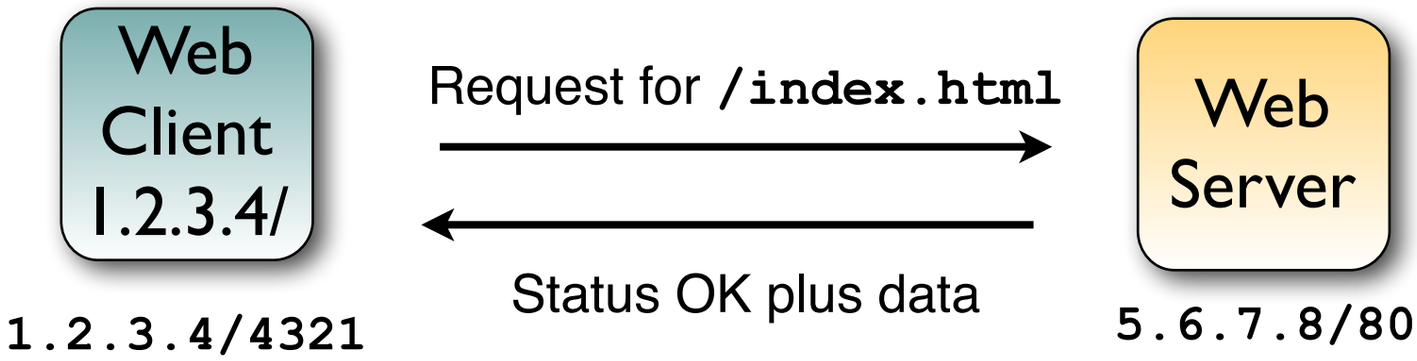
Stream of TCP packets

Event → `connection_established(1.2.3.4/4321⇒5.6.7.8/80)`

TCP stream reassembly for originator

Event → `http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`

Event Model



Stream of TCP packets

Event → **`connection_established(1.2.3.4/4321⇒5.6.7.8/80)`**

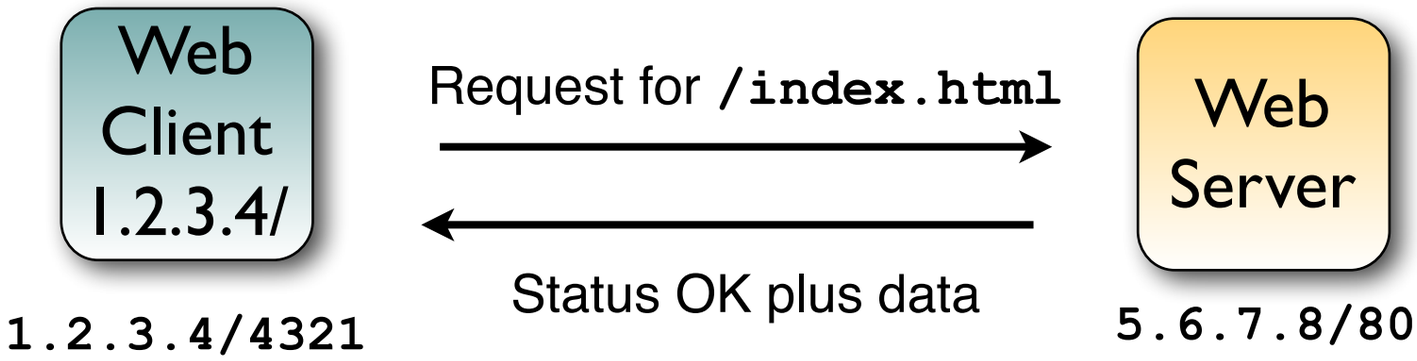
TCP stream reassembly for originator

Event → **`http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`**

TCP stream reassembly for responder

Event → **`http_reply(1.2.3.4/4321⇒5.6.7.8/80, 200, "OK", data)`**

Event Model



Stream of TCP packets

Event → **`connection_established(1.2.3.4/4321⇒5.6.7.8/80)`**

TCP stream reassembly for originator

Event → **`http_request(1.2.3.4/4321⇒5.6.7.8/80, "GET", "/index.html")`**

TCP stream reassembly for responder

Event → **`http_reply(1.2.3.4/4321⇒5.6.7.8/80, 200, "OK", data)`**

Event → **`connection_finished(1.2.3.4/4321, 5.6.7.8/80)`**